

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE MÓDULOS DE APLICATIVOS MÓVILES DE SISTEMA ERP UTILIZANDO PRÁCTICAS DEVOPS SEGURAS PARA MANTICORE LABS

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

RODMAN XAVIER IÑIGUEZ SARANGO

rodman.iniguez@epn.edu.ec

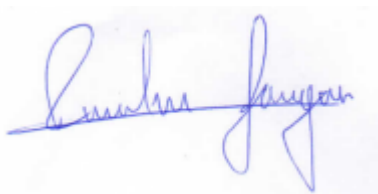
DIRECTOR: VICENTE ADRIÁN EGÜEZ SARZOSA

adrian.eguez@epn.edu.ec

DMQ, marzo 2023

CERTIFICACIONES

Yo, RODMAN XAVIER IÑIGUEZ SARANGO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado paraningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



RODMAN XAVIER IÑIGUEZ SARANGO

Certifico que el presente trabajo de integración curricular fue desarrollado por RODMAN XAVIER IÑIGUEZ SARANGO, bajo mi supervisión.



VICENTE ADRIÁN EGÜEZ
SARZOSADIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

RODMAN XAVIER IÑIGUEZ SARANGO
VICENTE ADRIÁN EGÜEZ SARZOSA

DEDICATORIA

El presente trabajo de titulación representa la culminación de una etapa importante, complicada y laboriosa de mi preparación intelectual con el objetivo de convertirme en una persona capaz, competente, innovador y lleno de otras cualidades que me permitirán aportar, con integridad y responsabilidad, ideas y/u opiniones a la expectativa de un profesional, realizar mi trabajo con ética y hostiad.

Con lo mencionado quiero dedicar este trabajo a mi familia, especialmente a mis padres que supieron brindarme todo su apoyo y aliento en los buenos y malos momentos, sobre todo en los momentos más complicados donde parecería todo ir en picada, ahí fue cuando más estuvieron para mí y supieron no dejarme caer, ahora próximo a graduarme quiero ofrecer este logro a ellos y demostrar de lo que soy capaz con la preparación que he recibido de los mejores maestros de la Escuela Politécnica Nacional quienes me brindaron una preparación de calidad durante todo el tiempo requerido para la culminación de mis estudios universitarios.

Rodman Xavier Iñiguez Sarango

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	II
DECLARACIÓN DE AUTORÍA.....	III
DEDICATORIA.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE TABLAS.....	VII
ÍNDICE DE FIGURAS.....	VIII
RESUMEN.....	IX
ABSTRACT.....	X
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	11
1.1 Objetivo general.....	11
1.2 Objetivos específicos.....	11
1.3 Alcance.....	12
1.4 Marco teórico.....	12
1.4.1 Enterprise Resource Planning (ERP).....	12
1.4.2 DevOps.....	13
1.4.3 Scrum.....	13
1.5 Herramientas de Desarrollo.....	15
1.5.1 Android Studio.....	15
1.5.2 Firebase.....	16
1.5.3 Figma.....	16
1.5.4 Git.....	17
1.5.5 GitKraken.....	18
1.5.6 GitLab.....	18
1.6 Lenguajes de Programación.....	18
1.6.1 Kotlin.....	18
1.6.2 YAML.....	19
2 METODOLOGIA.....	20
2.1 Implementación Scrum.....	20
2.2 Implementación DevOps.....	22
2.2.1 Configuración DevOps.....	22
2.2.2 Ejecución DevOps.....	26

2.3	Desarrollo del producto	28
2.3.1	Requerimientos e historias de usuario.....	29
2.3.1.1	Historia de usuario 01	29
2.3.1.2	Historia de usuario 02	29
2.3.1.3	Historia de usuario 03	30
2.3.1.4	Historia de usuario 04	30
2.3.2	Diseño.....	30
2.3.2.1	Modelo de Interfaces.....	31
2.3.2.2	Estructura de la colección de datos.....	38
2.3.3	Codificación y pruebas de funcionalidad	39
2.3.3.1	Sprint 01	39
2.3.3.2	Sprint 02	40
2.3.3.3	Sprint 03	43
2.3.3.4	Sprint 04	50
2.3.3.5	Sprint 05	53
2.3.4	Pruebas de Usabilidad	55
2.3.5	Despliegue	63
3	RESULTADOS	64
4	CONCLUSIONES Y RECOMENDACIONES	67
4.1	Conclusiones.....	67
4.2	Recomendaciones.....	68
5	REFERENCIAS BIBLIOGRÁFICAS.....	69
6	ANEXOS	71

ÍNDICE DE TABLAS

Tabla 1 Roles Scrum	20
Tabla 2 Artefactos Scrum	21
Tabla 3 Eventos Scrum.....	21
Tabla 4 Historia de usuario 01 - Resumen	29
Tabla 5 Historia de usuario 02 - Resumen	29
Tabla 6 Historia de usuario 03 - Resumen	30
Tabla 7 Historia de usuario 04 - Resumen	30
Tabla 8 Resultados Sprint review 01.....	40
Tabla 9 Resultados del Sprint 01	40
Tabla 10 Resultados Sprint review 02.....	42
Tabla 11 Resultados del Sprint 02	43
Tabla 12 Resultados Sprint review 03.....	49
Tabla 13 Resultados Sprint 03.....	49
Tabla 14 Resultados Sprint review 04.....	52
Tabla 15 Resultados Sprint 04.....	53
Tabla 16 Resultados Sprint review 05.....	54
Tabla 17 Resultados Sprint 05.....	54
Tabla 18 Preguntas SUS [20]	55
Tabla 19 Resultados SUS.....	56
Tabla 20 Equivalencia SUS	57
Tabla 21 Respuestas usuario 01	58
Tabla 22 Respuestas usuario 02	58
Tabla 23 Respuestas usuario 03	59
Tabla 24 Respuestas usuario 04	59
Tabla 25 Respuestas usuario 05	60
Tabla 26 Respuestas usuario 06	60
Tabla 27 Respuestas usuario 07	61
Tabla 28 Respuestas usuario 08	61
Tabla 29 Respuestas usuario 09	62
Tabla 30 Respuestas usuario 10	62
Tabla 31 Resultados pipelines	64
Tabla 32 Porcentajes de éxito por Sprint	65
Tabla 33 Resultados SUS.....	66

ÍNDICE DE FIGURAS

Figura 1 Flujo DevOps.....	22
Figura 2 Servicio Runner de GitLab.....	23
Figura 3 Vinculación Firebase con Google Cloud	24
Figura 4 Archivo Json de credenciales	24
Figura 5 Archivo Json ubicado en el repositorio remoto.....	25
Figura 6 Código de archivo yml para ejecución DevOps.....	25
Figura 7 Diagrama de trabajo DevOps.....	26
Figura 8 Commit repositorio local.....	26
Figura 9 Commit repositorio remoto.....	27
Figura 10 Estados de los pipelines ejecutados	27
Figura 11 Estado fallido de un pipeline	28
Figura 12 Vista detallada del pipeline con error	28
Figura 13 Interfaces gráficas – vista general.....	31
Figura 14 Pantalla - Inicio	32
Figura 15 Pantalla – Menú principal.....	33
Figura 16 Pantalla – Registrar contrato.....	34
Figura 17 Pantalla – Actualizar contrato.....	35
Figura 18 Pantalla – Rol de pagos.....	36
Figura 19 Pantalla – Buscar rol de pagos	37
Figura 20 Pantalla – Información rol de pagos	37
Figura 21 Estructura de colección de datos	38
Figura 22 Codificación archivo yml	41
Figura 23 Implementación servicio runner de GitLab	42
Figura 24 Commits GitKraken.....	44
Figura 25 Commits GitLab	44
Figura 26 Implementación pantalla - inicio	45
Figura 27 Implementación pantalla - menú principal	46
Figura 28 Implementación pantalla - registrar contrato	47
Figura 29 Implementación pantalla - actualizar contrato	48
Figura 30 Ejecución de flujo DevOps	48
Figura 31 Implementación pantalla - rol de pagos.....	51
Figura 32 Implementación pantalla - buscar rol de pagos	52
Figura 33 Implementación pantalla - ver rol de pagos.....	52
Figura 34 Ejecución de flujo DevOps final	53
Figura 35 Lanzamiento del producto por App-Distribution.....	54
Figura 36 Cuestionario SUS	57
Figura 37 App-Distribution lanzamiento del producto final	63
Figura 38 Flujos DevOps ejecutados	65

RESUMEN

El desarrollo y entrega de nuevas aplicaciones móviles se ha convertido en una carrera donde grandes y pequeñas empresas participan, para lograrlo implementan métodos, herramientas, innovaciones tecnológicas, metodologías, e inclusive inversión en personal, por ello, el documento a continuación analizará y pondrá en práctica un método muy utilizado por la mayoría llamado DevOps, cuyo modo de operar se enfoca en la integración continua de incrementos funcionales y entregas igualmente continuas de versiones utilizables del producto al cliente. Para lograr el propósito de examinar si las practicas DevOps son seguras y su rendimiento en producción son efectivos, se utilizará la metodología Scrum como apoyo para el desarrollo de un aplicativo móvil enfocado en el módulo de Rol de pagos de un sistema ERP, por lo cual durante el ciclo de vida se ejecutará algunas veces el flujo de trabajo DevOps para compararlo con una situación similar de desarrollo, pero sin el flujo de trabajo mencionado. Así se llegará a una conclusión sobre el método verificando si cumple con lo planteado inicialmente o caso contrario no lo logra.

PALABRAS CLAVE: Flujo DevOps, Integración Continua, DespliegueContinuo, Rol de pagos, Android.

ABSTRACT

The development and delivery of new mobile applications has become a race where large and small companies participate, to achieve this they implement methods, tools, technological innovations, methodologies, and even investment in personnel, therefore, the document below will analyze and put into practice a method widely used by the majority called DevOps, whose mode of operation focuses on the continuous integration of functional increments and equally continuous delivery of usable versions of the product to the client. To achieve the purpose of examining whether DevOps practices are safe and their performance in production are effective, the Scrum methodology will be used as support for the development of a mobile application focused on the Payment Role module of an ERP system, for which During the life cycle, the DevOps workflow will be executed a few times to compare it with a similar development situation, but without the mentioned workflow. In this way, a conclusion will be reached about the method, verifying if it complies with what was initially proposed or, if not, it does not.

KEYWORDS: DevOps flow, Continuous Integration, Continuous Deployment, Payroll, Android.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Los sistemas ERP conocidos por sus siglas en inglés Enterprise Resource Planning, son sistemas que ayudan a las empresas a brindar servicios para sus clientes externos, internos y proveedores, siendo herramientas para la gestión de toda una empresa.

Los ERP existen ya varios años en el mercado, pero siempre están en constante evolución, no solamente en la arquitectura sino también en el desarrollo de funcionalidades nuevas o mejoras de estos.

Las prácticas DevOps permiten que se implementen flujos automáticos para el despliegue en diferentes ambientes, ahorrando mucho tiempo y permitiendo que el flujo de pruebas, desarrollo, implementación y despliegue se lleve a cabo fácilmente. Hay que tomar en cuenta la seguridad ya que se manejan varias claves y credenciales durante este proceso.

Así comprendido brevemente los sistemas ERP el presente trabajo se enfoca en el desarrollo de un componente vital como lo es el módulo de Rol de pagos para la empresa Manticore Labs. El componente estará comprendido por varias funciones como lo son:

- Registro de contratos nuevos.
- Actualización de los contratos.
- Generador de rol de pagos.
- Buscador y visualizador de los roles de pagos, vista general y detallada.

1.1 Objetivo general

Implementar prácticas DevOps seguras para el desarrollo de módulos en aplicativos móviles de sistemas ERP para la empresa Manticore Labs.

1.2 Objetivos específicos

1. Definir prácticas DevOps seguras para el desarrollo de un aplicativo móvil.
2. Implementar y probar un aplicativo móvil de sistema ERP enfocado en el módulo de Roles de Pago utilizando una práctica DevOps segura.
3. Comparar el desarrollo del aplicativo móvil con prácticas DevOps seguras y sin prácticas DevOps seguras.

1.3 Alcance

El alcance del proyecto es el levantamiento de ambientes con prácticas DevOps seguras y la implementación de un aplicativo móvil de un sistema ERP para el módulo de Roles de Pago en la empresa Manticore Labs, utilizando las metodologías ágiles partiendo de la planificación para definir los requisitos y diseño del sistema; la codificación donde se realizará la implementación del código del producto y las pruebas para garantizar la eficiencia y calidad del mismo.

1.4 Marco teórico

Con el crecimiento exponencial de la tecnología haciendo uso de nuevas técnicas y materiales, junto con el alto nivel de competencia en la calidad, en la velocidad del desarrollo, en la inventiva de nuevas funciones o servicios, las pequeñas, medianas y grandes empresas de tecnología han creado e implementados diferentes métodos para la producción de hardware o software en base al objetivo de cada una de ellas. En el presente trabajo se expondrá un método conocido como DevOps para el desarrollo de software, tomando como enfoque una aplicación móvil centrada en un sistema ERP orientado al módulo de gestión de los roles de pagos. De igual manera por la aplicación del método ya mencionado se ahondará en una metodología enfocada en etapas para el ciclo de vida del desarrollo de la aplicación, conocida como SCRUM, ayudando a la gestión de recursos, tiempo y asegurando un nivel de calidad óptimo.

1.4.1 Enterprise Resource Planning (ERP)

Con la evolución tecnológica, la demanda del mercado y la creación de nichos de uso de productos o servicios, las empresas se han visto en la necesidad de innovarse y/o expandirse con nuevas áreas de trabajo internas y en algunos casos externas, volviendo un poco complicado la administración y gestión de cada una de las áreas, las cuales en décadas pasadas de manejaban de manera rudimentaria. A partir de ellos se buscó optimizar estos procesos dando origen a una planificación conocida como ERP.

La Planificación de Recursos Empresariales, o ERP (Enterprise Resource Planning) se enfoca en sistemas de software empleados para la administración de procesos

claves empresariales en las finanzas, en los servicios, durante la manufactura, entre otros. En otras palabras, un sistema ERP se puede utilizar para simplificar y automatizar procesos importantes permitiendo tener un mejor rendimiento en la administración y gestión de las diferentes áreas de la empresa al igual que reducir tiempos de comunicación entre ellas optimizando la productividad [1].

1.4.2 DevOps

DevOps, es un término cuyo origen se encuentra en la combinación de development (desarrollo) y operations (operaciones). Una filosofía de trabajo que aspira a un mejor desarrollo de aplicaciones en menos tiempo así ofreciendo nuevas y funcionales operaciones de un software o producto a los clientes [2].

Un DevOps abarca todo el ciclo de vida, desde planificación inicial pasando por las diferentes fases de codificación, pruebas, publicación, incluida una supervisión continua si el caso lo amerita [3].

Existen diferentes métodos DevOps que las empresas instauran en su modo de trabajo buscando mejorar el desarrollo de aplicaciones y acelerar las publicaciones de nuevos productos al igual que mejoras de estos. Entre los más populares están Scrum, Kanban y Agile [3].

1.4.3 Scrum

Scrum define la manera en que los diferentes equipos de trabajo de un proyecto deben colaborar para culminar un producto y asegurar la calidad de este. Las practicas Scrum plantean crear flujos de trabajo específicos llamados Sprint con el objetivo de entregar avances concretos en un periodo de tiempo establecido, también propone crear roles específicos buscando una comunicación continua entre las diferentes áreas y roles para asegurar un entendimiento optimo en la funcionalidad al igual de que la calidad del producto [3].

La planificación Scrum cuenta con tres aspectos:

- Roles
- Eventos
- Artefactos

Roles

Para la ejecución del Scrum se necesitan roles con funciones específicas, dichos roles son:

- Product Owner: Es o son los interesados en la culminación del producto con calidad y todas sus operaciones funcionales, se podría asegurar que en el cliente.
- Scrum Master: Es el encargado de dirigir y gestionar el proyecto.
- Desarrollador: Es el equipo de trabajo enfocado en el diseño, la codificación, realización de pruebas y mantenimiento de la aplicación de ser necesario [4].

Eventos

Los eventos son reuniones planificadas con diferentes periodos de tiempo, pero con el mismo objetivo, la culminación del producto en desarrollo, dichos eventos son:

- Sprint: Cuenta con el periodo de tiempo más largo con un aproximado de 4 semanas en el cual se abarca a los de eventos.
- Sprint Planning: Es donde se planifica cada uno de los Sprint, estableciendo las actividades a realizarse.
- Daily Scrum: Es la reunión diaria conformada por el Scrum Master y el equipo de desarrollo para evaluar los resultados obtenidos durante el Sprint al igual de los impedimentos u obstáculos que han surgido y así plantear soluciones rápidas y eficaces.
- Sprint Review: Se enfoca en evaluar el progreso al final de cada Sprint teniendo como objetivo cumplir con lo establecido en el Sprint Planning.
- Sprint Retrospective: Su objetivo es evaluar el desempeño del equipo de trabajo y plantear soluciones administrativas de encontrarse algún

problema [4].

Artefactos

Los artefactos comprenden los documentos o estados registrados del desarrollo del producto, se definen tres artefactos:

- Product Backlog: Esta conformado por los requerimientos solicitados por el cliente para el desarrollo del producto.
- Sprint Backlog: Es el conjunto de requerimientos propuestos a cumplir dentro del periodo de tiempo de cada Sprint con el propósito de hacer entrega de funciones parciales con un peso de valor significativo.
- Incremento: Es a entrega del producto parcialmente finalizado, pero con valor significativo en un punto o puntos cualquiera durante el ciclo de vida de desarrollo del producto [4].

1.5 Herramientas de Desarrollo

1.5.1 Android Studio

El desarrollo de aplicaciones de software ha tenido grandes y diferentes cambios desde sus comienzos con las tarjetas perforadas, pasando por aplicaciones nativas en los ordenadores hasta llegar al desarrollo de aplicaciones móviles para teléfonos y aplicaciones web para la red, estos cambios han traído diferentes herramientas como lo son los entornos de desarrollo integrado, o IDE (Integrated Development Environment).

Entre los diferentes IDEs para el desarrollo de aplicaciones móviles en sistemas Android conocidos actualmente existe Android Studio, un entorno de desarrollo oficial para sistemas Android basado en un editor de código muy eficaz y contando con diferentes herramientas útiles para desarrolladores de IntelliJ IDEA (IDE enfocado en lenguajes de programación Java y Kotlin) [5].

1.5.2 Firebase

Google dispone de un gran catálogo de herramientas para usuarios comunes y usuarios más experimentados como son los desarrolladores, para ellos tiene a su disposición herramientas especializadas vinculadas con el sistema operativo Android, una de estas herramientas es Firebase.

Firebase es una plataforma móvil que brinda a los usuarios experimentados y poco experimentados varias herramientas con diferentes utilidades para el desarrollo de aplicaciones web y móviles de manera eficiente con gran calidad mientras se hace todo de manera rápida, buscando optimizar el rendimiento de estas [6].

Dos herramientas que se usarán para el desarrollo del producto enfocado en este documento serán:

- **Cloud Firestore**

Con Cloud Firestore se pueden crear base de datos NoSQL, es decir no se necesita una estructura para almacenar y relacionar la información. Firestore utiliza colecciones de documentos que representa las entidades en los cuales se almacena la información que el desarrollador empleara en el uso de la aplicación [7].

- **Firebase App-Distribution**

App-Distribution ofrece al desarrollador distribuir y lanzar el producto realizado de una forma más fácil, es decir, y tomando como ejemplo en proyecto base del presente documento, se realizará un lanzamiento automático hacia un repositorio de un producto terminado y ejecutable [8].

1.5.3 Figma

Es una herramienta web que ofrece la posibilidad de realizar prototipos web y móviles de manera gráfica ofreciendo diferentes herramientas con un gran catálogo de figuras, funcionalidad, colores, entre otros, al igual de permitir al usuario dueño del prototipo compartir su trabajo y volverlo colaborativo o compartirlo a manera de presentación, pero con una ligera funcionabilidad dando

un mayor aporte al prototipo desarrollado sobre como seria el producto a desarrollar [9].

1.5.4 Git

Las empresas con sus diferentes proyectos en desarrollo de software se han visto en la necesidad de poder controlar las versiones de estos, es decir, gestionar versiones hasta puntos seguros evitando así perder los avances logrados y teniendo que volver a empezar. Para lograr este objetivo han implementado Controladores de Versiones con la capacidad de registrar y almacenar un historial de los cambios realizados sobre los archivos del proyecto.

Así la herramienta Git a viene a ser un sistema de control de versiones permitiendo gestionar cada versión de forma local en un computador al igual que de forma remota por medio de líneas de comando o utilizando interfaces graficas como GitKreaken [10].

Git cuenta con tres partes para su funcionamiento, las cuales son:

- Repositorios: Sirve como un espacio de almacenamiento a manera de carpeta donde se encuentran los archivos del proyecto del cual se llevarán un historial de versiones [11].
- Ramas: También conocidas como líneas de trabajo, representan diferentes entornos de trabajo de un mismo proyecto en desarrollo, todo al mismo tiempo, dedicadas a diferentes funcionalidades o fines [12].
- Operaciones: Las operaciones básicas de la herramienta Git para trabajos remotos constan de; operación Pull, permitiendo al usuario traer los archivos del repositorio remoto de las diferentes ramas según corresponda o necesite al repositorio local; operación Push, permitiendo de manera inversa al Pull enviar archivos modificados al repositorio remoto a la rama correspondiente y finalmente la operación Commit, captura los cambios efectuados ese instante del proyecto creando las versiones seguras [13].

1.5.5 GitKraken

GitKraken es una herramienta multi plataforma que permite al usuario operar Git con mayor facilidad brindando una interfaz gráfica amigable y fácil de comprender, así evitando el uso de comandos por parte del Git. Esta herramienta permite vincularse con plataformas de repositorios como lo son GitHub, GitLab entre otras ofreciendo facilidades para el desarrollo de proyectos con el control de sus versiones, monitoreo de los mismo y un tablero para dar seguimiento a las actividades a realizar [14].

1.5.6 GitLab

GitLab es una plataforma de almacenamiento que permite a los diferentes proyectos contar con un repositorio el cual se puede administrar, crear y conectar con diferentes aplicaciones para realizar todo tipo de integración de las diferentes etapas de un DevOps [15].

CI/CD

Gitlab CI/CD proviene de los términos Integración Continua y Entrega Continua vistos con anterioridad, esta herramienta de GitLab permite al usuario efectuar dichos procesos de manera automática facilitando procesos como la construcción, las pruebas, el lanzamiento y el monitoreo de cada entrega del producto en construcción en base al DevOps [16].

Runners

Los procesos realizados por el CI/CD de manera automática necesitan un entorno o sistema para ejecutarse, aquí es donde los Runners se hacen presente y permiten ejecutar los procesos en máquinas virtuales, servicio ofrecido por GitLab, o máquinas locales utilizando un contenedor [17].

1.6 Lenguajes de Programación

1.6.1 Kotlin

El lenguaje de programación Kotlin es un lenguaje de código abierto que permite la programación funcional y la programación orientada a objetos para el

desarrollo de aplicaciones móviles, Kotlin posee una sintaxis muy parecida a la de otros lenguajes como C# y Java con la ventaja que brinda algunas facilidades durante la codificación volviéndolo un lenguaje práctico y comprensible [18].

1.6.2 YAML

El lenguaje de programación YAML (Yet Another Markup), comprende un lenguaje de serialización de datos, con la finalidad de configurar los diferentes entornos o sistemas capaces de entender la sintaxis, así esta configuración establecida por parte del usuario será capaz de preparar un entorno para la ejecución de los requerimientos, las opciones y los comandos necesarios para el desarrollo del proyecto [19]. En el presente proyecto en desarrollo se empleará para la configuración del Runner, parte fundamental del método DevOps.

2 METODOLOGIA

En el desarrollo del presente proyecto para la construcción de un aplicativo móvil de un sistema ERP enfocado en el módulo de Rol de Pagos, se contemplaron e implementaron dos conceptos, vistos a detalle en el capítulo anterior, el primero llamado DevOps enfocado en la integración continua del aplicativo móvil haciendo de uso de herramientas como GitLab, y el segundo, la metodología SCRUM con el cual se planteará el ciclo de vida a seguir para el desarrollo del producto.

Ya establecida la forma de trabajo fue preciso definir la implementación de cada una de las metodologías mencionadas ya que cuentan con sus respectivas configuración y etapas a seguir que se presentarán en las secciones siguientes.

2.1 Implementación Scrum

La metodología Scrum fue diseñada para trabajar con diferentes equipos de codificación, por lo cual para el desarrollo del módulo rol de pagos del proyecto, con un miembro en el equipo de codificación presentará ligeras modificaciones para cumplir con el objetivo.

Como se presentó en la sección 1.4.3, Scrum cuenta con diferentes aspectos los cuales son roles, evento y artefactos, de tal manera lo primero en realizarse fue en establecer los roles, los cuales se presentan en la Tabla 1.

Tabla 1 Roles Scrum

Rol	Persona	Función
Scrum Master	Adrián Egüez	Presentado como el director del proyecto, encargado de definir los avances a efectuar del producto y su respectiva evaluación.
Product Owner	Adrián Egüez	Ocupando un puesto de revisor, se encarga que las entregas parciales funcionales del producto cumplan con lo requerido al igual que la entrega final tengo el valor solicitado por el cliente.

Desarrollador	Rodman Iñiguez	Miembro del equipo de codificación encargado de los procesos de producción durante el de ciclo de vida del producto.
----------------------	----------------	----------------------------------------------------------------------------------------------------------------------

Establecidos los roles con sus responsabilidades y quien los desempeñará lo siguiente fue definir los artefactos, en la Tabla 2. se puede observar una descripción de cada uno de ellos.

Tabla 2 Artefactos Scrum

Artefacto	Descripción
Product Backlog	Consta de las historias de usuarios brindando toda la información posible sobre el diseño y funcionalidad del aplicativo permitiendo comprender al desarrollador los requerimientos a codificar.
Sprint Backlog	Aquí las historias de usuarios serán divididas en bloques y cada bloque asignado a un Sprint, el objetivo es cumplir con incrementos de valor en cada iteración.
Incremento	Son entregas parciales del producto final, pero con funcionalidad aportando valor con cada incremento hasta la entrega final del aplicativo.

Finalmente se estableció los eventos Scrum, a diferencia de lo expuesto en la sección 1.4.3 aquí fue donde se hicieron algunos cambios los cuales fueron tomar solo 3 eventos de los 5 que son en total, este cambio se realizó debió al equipo de trabajo que consta de una sola persona con el rol de desarrollador, así en la Tabla 3 se puede observar cada evento Scrum utilizado junto con su descripción sobre su papel en el proyecto.

Tabla 3 Eventos Scrum

Artefacto	Descripción
Sprint	Para el desarrollo del módulo rol de pagos se establecieron 5 sprint con una duración de una semana cada uno, el objetivo durante el tiempo marcado es cumplir con el ciclo de vida del desarrollo del aplicativo móvil.
Sprint Planning	En cada sprint planning se detallaron los objetivos a cumplir para el sprint en ejecución.

Sprint Review	<p>Durante la reunión del sprint review se buscaba responder 3 preguntas.</p> <p>¿Qué se hizo en este Sprint?</p> <p>¿Qué inconvenientes encontré en este Sprint?</p> <p>¿Qué se hará en el siguiente Sprint?</p>
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.2 Implementación DevOps

Como el objetivo principal del presente proyecto es implementar practicas DevOps para el desarrollo de un aplicativo móvil, se utilizó un flujo DevOps que entrelaza las diferentes herramientas mencionadas en el marco teórico del presente documento, procurando obtener una buena práctica de desarrollo, la Ilustración 2.2.1 muestra el flujo a seguir.

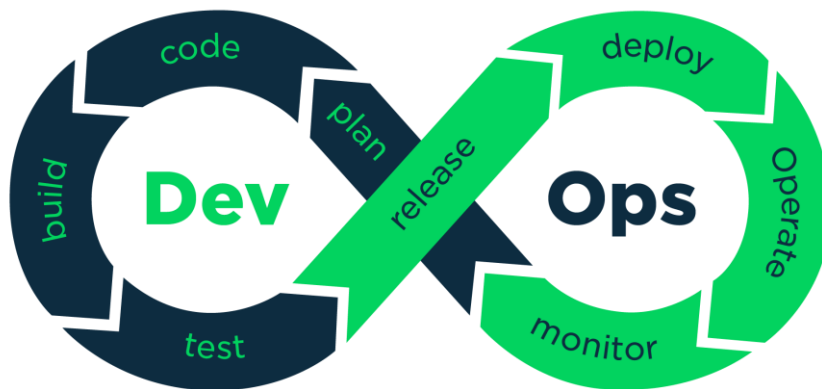


Figura 1 Flujo DevOps

Como se menciona en la sección 1.4.2, y se indica en la figura 2.2.1, DevOps hace referencia a un ciclo entre los procesos de desarrollo (dev) y los procesos de operaciones (ops) por medio de las herramientas utilizadas como Android Studio, Git, GitLab, por lo que, primeramente, se analizó las diferentes opciones de aplicación del método DevOps para cumplir con el flujo de trabajo referenciado.

2.2.1 Configuración DevOps

Con el análisis obtenido de los métodos DevOps, y disponiendo de las herramientas previamente instaladas y/o configuradas, según sea el caso, es momento de configurar el flujo de trabajo DevOps elegido para el desarrollo del presente proyecto.

Como se explicó en la sección 1.5.6 del marco teórico, los Runners son entornos con un sistema operativo que se encargara de ejecutar el flujo de trabajo DevOps dando como resultados si cada flujo ejecutado tuvo éxito, fue fallido o si tuvo que cancelado. Por lo que se verificó la disponibilidad y se procedió a la activación del servicio runner ofrecido por la plataforma GitLab, para ello, dentro del repositorio del proyecto en la plataforma GitLab en el panel lateral se debe ingresar a *setting >> CI/CD*, expandir el submenú Runners y activar la parte de 'Shared runners', como se muestra en la figura 2.

De esta manera el desarrollador puede evitar instalar un entorno de forma manual para ejecutar los flujos de trabajo y tener en la plataforma GitLab de manera íntegra y más controlada todo el proceso.

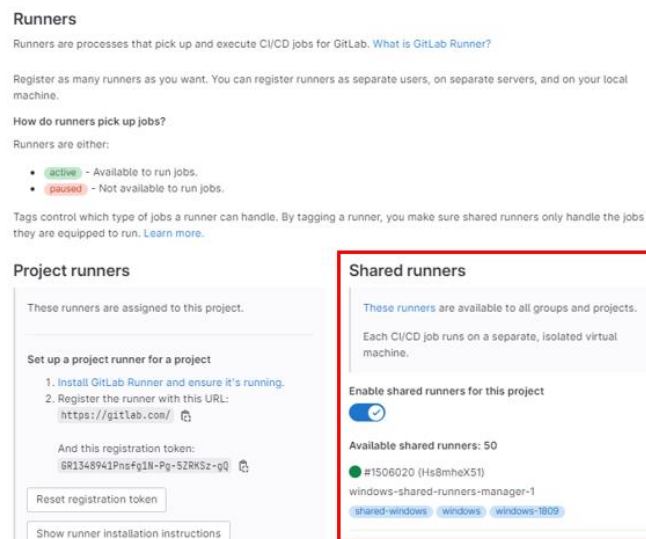


Figura 2 Servicio Runner de GitLab

Una vez teniendo disponible el entorno de trabajo es momento de vincular el proyecto ubicado en Firebase con Google Cloud, figura 3, cuyo objetivo es crear las credenciales por medio de un archivo Json generado automáticamente, Figura 4 que permitirá ejecutar el entorno Runner para las pruebas de la aplicación y usar App-Distribution para los lanzamientos con incrementos de valor en las diferentes versiones ejecutables del producto.

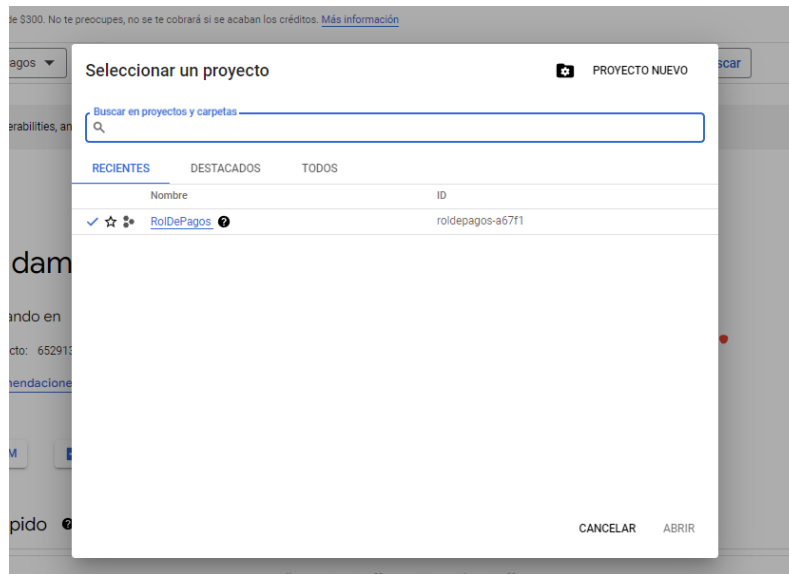


Figura 3 Vinculación Firebase con Google Cloud

DETALLES PERMISOS **CLAVES** MÉTRICAS REGISTROS

Claves

⚠ Las claves de cuenta de servicio podrían poner en riesgo la seguridad si se ven comprometidas. Te recomendamos que no descargues claves de cuenta de servicio. [Federación de Workload Identity](#). Puedes obtener más información sobre cuál es la mejor manera de autenticar las cuentas de servicio en Google Cloud [aquí](#).

Agrega un nuevo par de claves o sube un certificado de clave pública de un par de claves existente.

Impide la creación de claves de cuentas de servicio con las [políticas de la organización](#).

[Más información para configurar políticas de la organización en cuentas de servicio](#)

AGREGAR CLAVE ▾

Tipo	Estado	Clave	Fecha de creación de la clave	Fecha de vencimiento de la clave	
🔑	Activa	af043318f2c45dcf0ca05ba032979abed837a03e	12 feb 2023	31 dic 9999	🗑

Figura 4 Archivo Json de credenciales

Un punto importante para no pasar por alto es, el archivo Json creado con las credenciales debe estar en alguna parte dentro del repositorio en la plataforma GitLab para que se ejecute correctamente el proceso, figura 5, caso contrario el entorno no podrá ejecutarse dando como resultado fallido en cada flujo de trabajo DevOps que el desarrollador lance.

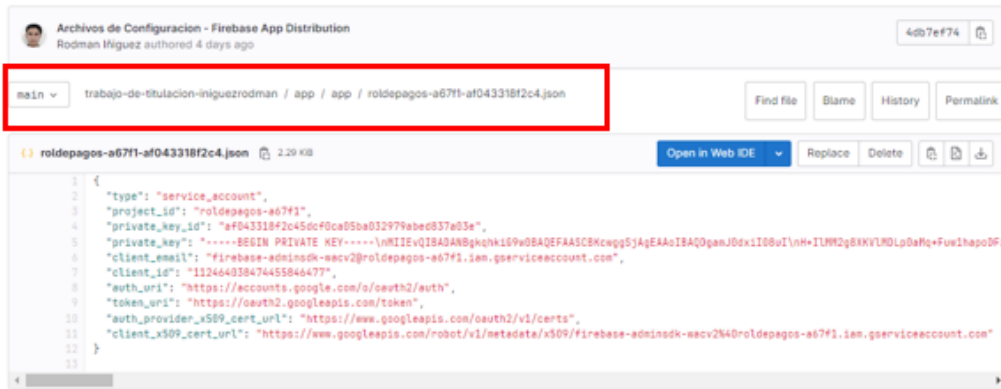


Figura 5 Archivo Json ubicado en el repositorio remoto

Finalmente es momento de crear un archivo tipo .yml, Este archivo contendrá un código, figura 6, que ejecutará el flujo DevOps en sí, es decir se encargará de ejecutar la integración de cada incremento con valor para la rama de trabajo Development y la compilación dentro de la rama de trabajo Deploy, donde al finalizado el flujo de manera exitosa se creará un APK posteriormente almacenado en App-Distribution para sus pruebas respectivas por el personal correspondiente (según un esquema de trabajo completo contando con varios equipos para el desarrollo).

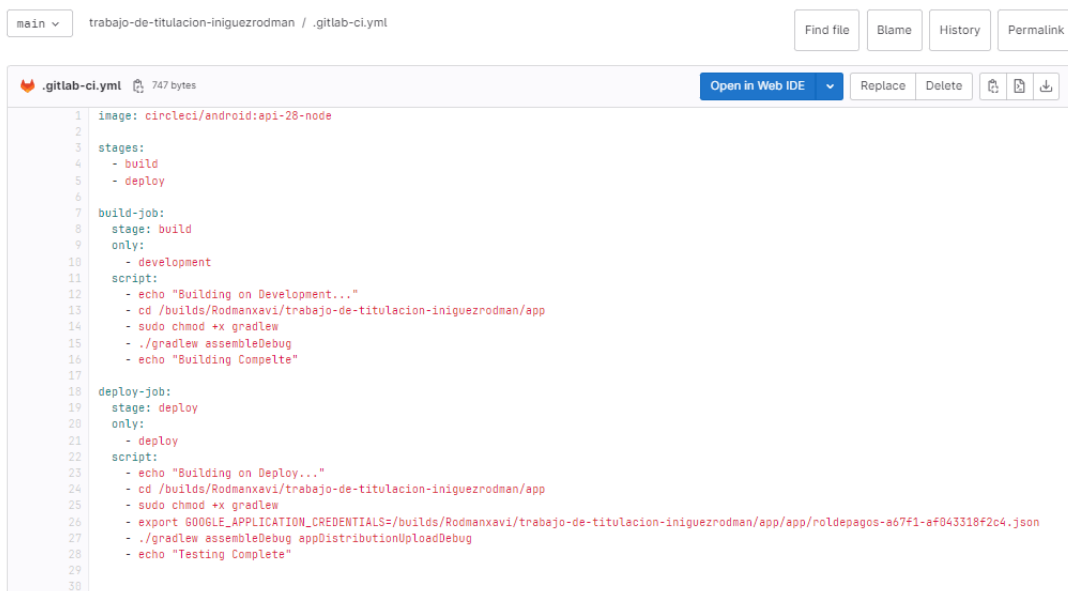


Figura 6 Código de archivo yml para ejecución DevOps

2.2.2 Ejecución DevOps

Con las configuraciones correspondientes realizadas y las herramientas de trabajo conectadas entre sí, surge un diagrama del proceso a seguir para cumplir con la ejecución establecida y cumplir con el DevOps, dicho diagrama se muestra en la figura 7.

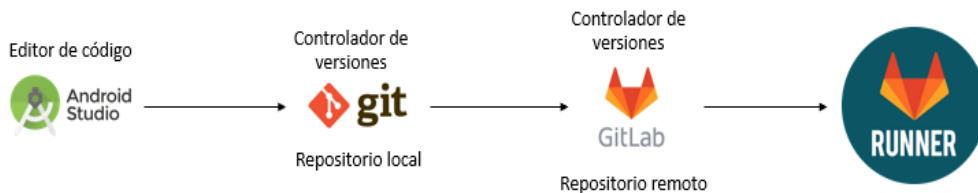


Figura 7 Diagrama de trabajo DevOps

El proceso más complejo ya fue culminado, la configuración de las diferentes plataformas y herramientas al igual que su interconexión debe estar en total funcionamiento, así en la ejecución expuesta a continuación reflejara los resultados deseados.

En el IDE de programación Android Studio se realizarán las codificaciones correspondientes, al cumplir con un avance importante aportando funcionalidad adicional y relevante al producto se ejecutará un commit de manera local usando la herramienta GitKraken, figura 8, para guardar la versión actual del proyecto y posterior ejecutando un push al repositorio remoto, figura 9, empezando el flujo DevOps.

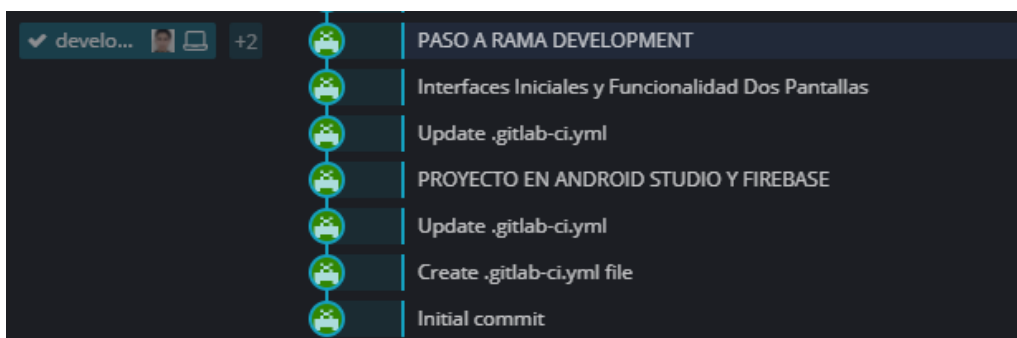


Figura 8 Commit repositorio local

Commit 30bc2eb6 authored 1 week ago by Rodman Iñiguez

Browse files

Options

PASO A RAMA DEVELOPMENT

Inclusión de pocas líneas, antes del paso al desarrollado de la aplicación en la rama Development

parent f7dcd20 main

No related merge requests found

Pipeline #774507021 passed with stage in 2 minutes and 32 seconds

Changes 2 Pipelines 4

Showing 2 changed files with 30 additions and 11 deletions

Hide whitespace changes

Inline

Side-by-side

```

app.idea/misc.xml
+1 -1
View file @30bc2eb6
...  ...  @@ -4,7 +4,7 @@
4 4  <option name="filePathToZoomLevelMap">
5 5  <map>
6 6  <entry key="..:/TIC_RolesDePago/trabajo-de-titulacion-
iniguezrodman/app/app/src/main/res/layout/activity_actualizar_contrato.xml" value="0.1" />
7 - <entry key="..:/TIC_RolesDePago/trabajo-de-titulacion-
iniguezrodman/app/app/src/main/res/layout/activity_crear_contrato.xml" value="0.1" />

```

Figura 9 Commit repositorio remoto

Ya efectuado los cambios relevantes se comprueba en la sección pipelines del menú CI/CD si el proceso se ejecutó con éxito u obtuvo alguna falla, como se observa en la figura 10, el proceso dentro de la rama Development transcurrió con éxito para posteriormente compilar dentro de la rama Deploy, como se configuró en el archivo yml, también transcurriendo con éxito y comprobando el método DevOps implemento funciona.

passed PASO A RAMA DEVELOPMENT #774507021 deploy 30bc2eb6 00:02:32 1 week ago

passed PASO A RAMA DEVELOPMENT #774506746 development 30bc2eb6 00:02:28 1 week ago

Figura 10 Estados de los pipelines ejecutados

A manera de ejemplo y en caso de un pipeline fallido como se muestra en la figura 11, dando un clic en el apartado de los 'Stages', se puede visualizar de manera más detalla el proceso para encontrar el error y solucionarlo.

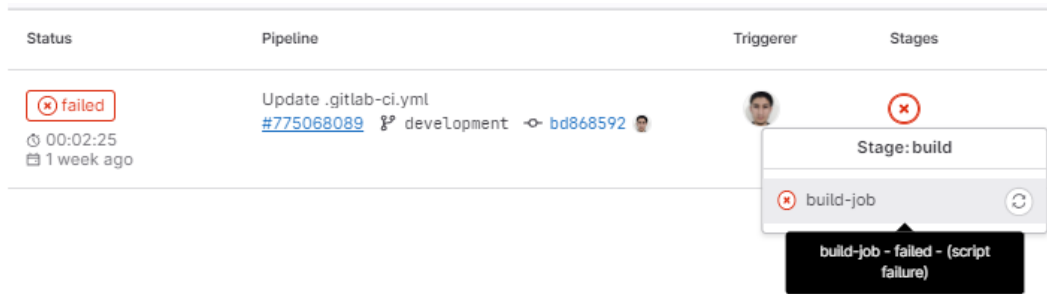


Figura 11 Estado fallido de un pipeline

Como se explicó en el punto anterior al ingresar ingresando al apartado de Stage se puede visualizar cada línea ejecutada por el entorno detallando el pipeline procesado, figura 12, brindando información extra para análisis y determinar el error correspondiente. Para este caso específico, durante la configuración del archivo yml la línea de comando describiendo la ubicación del archivo json con las credenciales de Google Cloud se encontraba con error provocando el estado de fallo en el flujo.

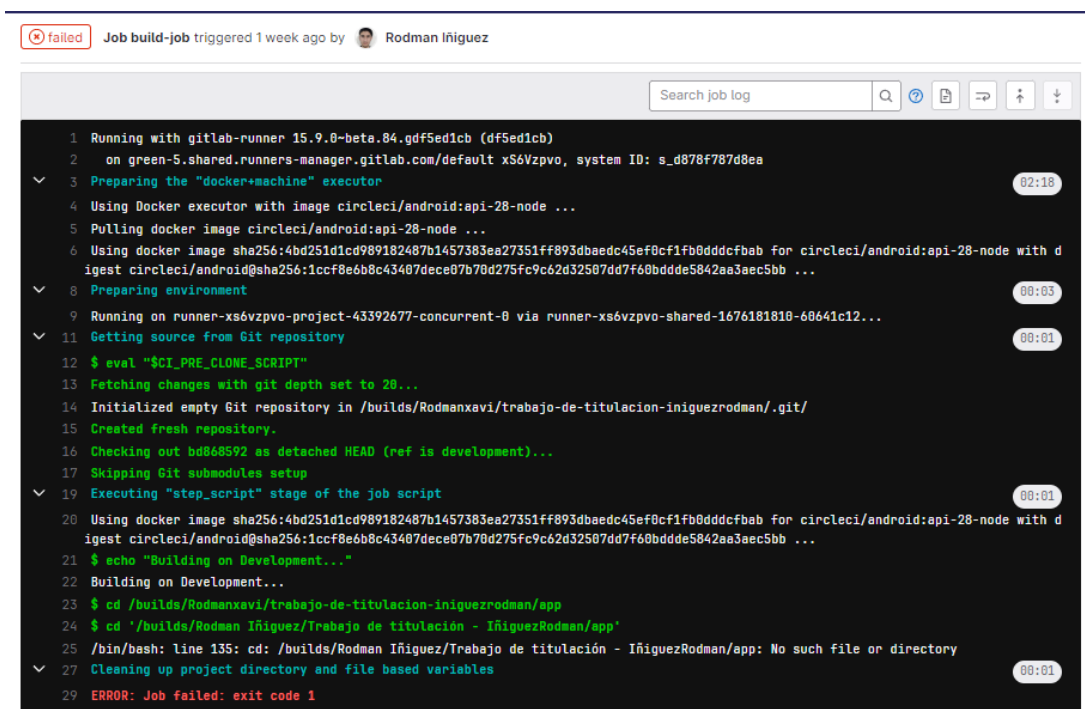


Figura 12 Vista detallada del pipeline con error

2.3 Desarrollo del producto

Realizada la configuración DevOps correspondiente y comprendida la metodología Scrum expuesta en la sección 1.4.3. se establecieron las siguientes fases para el desarrollo del producto:

- Requerimientos e historias de usuario.
- Diseño.
- Codificación y pruebas de funcionalidad.
- Pruebas de usabilidad.
- Despliegue.

2.3.1 Requerimientos e historias de usuario

Para la fase de requerimientos se realizó un análisis sobre las especificaciones y funcionalidades que debe tener el módulo de rol de pagos para cumplir con lo esperado, se contempló el diseño, la base de datos y funcionalidades con el objetivo de obtener toda la información posible y visualizar dicha información por medio de historias de usuarios. A continuación, se presentará a manera de resumen las historias de usuarios obtenidas con la información relevante para el desarrollo del proyecto.

2.3.1.1 Historia de usuario 01

Tabla 4 Historia de usuario 01 - Resumen

Descripción General
Permite la creación de nuevos contratos de trabajo registrando información del empleado.
Lista de Escenarios
Registro de contratos nuevos
Verificación de datos validos a registrar (Historia de usuario completa, Anexo II)

2.3.1.2 Historia de usuario 02

Tabla 5 Historia de usuario 02 - Resumen

Descripción General
Permite la modificación de contratos ya registrados en la base de datos.
Lista de Escenarios
Buscar contrato a modificar por medio de un parámetro.
Registro de datos nuevos

Verificación de datos validos a registrar (Historia de usuario completa, Anexo III)

2.3.1.3 Historia de usuario 03

Tabla 6 Historia de usuario 03 - Resumen

Descripción General
Permite generar roles de pagos vinculados a un contrato.
Lista de Escenarios
Buscar empleado a generar rol de pagos.
Visualizar datos relevantes del empleado.
Calcular sueldo a pagar

(Historia de usuario completa, Anexo IV)

2.3.1.4 Historia de usuario 04

Tabla 7 Historia de usuario 04 - Resumen

Descripción General
Permite buscar y visualizar los roles de pagos generados en vista resumida y detallada.
Lista de Escenarios
Visualizar lista de roles de pagos, vista resumida.
Buscar roles de pagos de un empleado.
Visualizar rol de pago, vista detallada.

(Historia de usuario completa, Anexo V)

2.3.2 Diseño

Una vez recopilado una gran cantidad de requerimientos y detalles por medio de las historias de usuarios se procede a realizar un modelado de las interfaces gráficas en base a la información previamente obtenida, permitiendo contemplar nuevos requerimientos importante que no eran visibles hasta este punto, de igual manera se definieron las reglas de negocio dando a conocer más requerimientos de funcionalidades atrás de las interfaces graficas.

Otro punto importante que se logró modelar, después de obtener los requerimientos de los diferentes métodos descritos anteriormente, fue la base de datos o para este caso en particular al ser un aplicativo móvil, se definió la

colección de datos que se empleara para el correcto funcionamiento del producto.

Así a continuación se presenta, en base a todos los requerimientos totales obtenidos, el diseño de las interfaces graficas a codificar y la estructura de la colección de datos.

2.3.2.1 Modelo de Interfaces

El aplicativo móvil de rolde pagos consta de 7 interfaces gráficas, las cuales disponen de todo lo necesario para crear roles de pagos asociados a un empleado por un tipo de contrato, cada pantalla diseñada cumple con un propósito específico para realizar el trabajo de manera fácil y comprensible, además contempla la posibilidad de efectuar cambios a futuro evitando lo más posible un retrabajo para efectuar dichos cambios, en la figura 13 se puede visualizar el diseño general de cada interfaz gráfica. (para revisar Mockup ver Anexo VII).

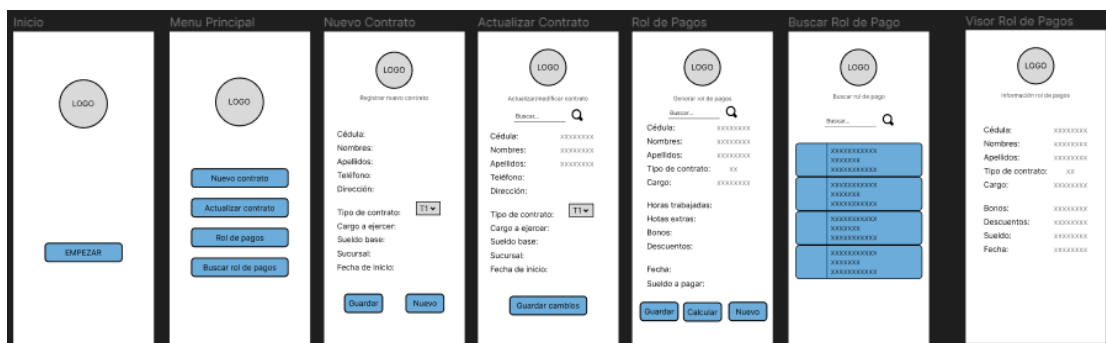


Figura 13 Interfaces gráficas – vista general

A continuación, se presentará cada pantalla diseñada junto con su funcionalidad.

La figura 14 muestra la pantalla de inicio del aplicativo móvil, una pantalla simple pensada para la integración del módulo rol de pagos al sistema ERP completo, esta pantalla presenta un solo botón que permite ingresar al menú principal.



Figura 14 Pantalla - Inicio

La siguiente pantalla ya contempla las reglas de negocio, aquí se puede observar el menú principal que dará paso al trabajo que necesite realizar el usuario, el menú cuenta de 4 opciones cada una dirige a una sección específica del aplicativo, dichas opciones son, y se pueden visualizar en la figura 15:

- Nuevo contrato.
- Actualizar contrato.
- Rol de pagos
- Buscar rol de pagos.



Figura 15 Pantalla – Menú principal

A continuación, la figura 16 muestra la primera pantalla de trabajo para el usuario al presionar el botón 'Nuevo contrato' del menú principal. En esta pantalla se podrán crear los contratos nuevos de los empleados, guardando información como: cedula, nombre, apellidos, teléfono dirección, tipo de contrato, cargo a ejercer, sueldo base y la fecha de inicio, esta información estará validada por lo cual al ingresar formatos errores en los campos y presionar el botón 'Guardar' se desplegará un aviso para realizar la corrección adecuada, caso contrario se almacenará la información en la colección de datos. También cuenta con un botón 'Nuevo' con el propósito de limpiar los campos y poder registrar un nuevo contrato.

LOGO

Registrar nuevo contrato

Cédula:
Nombres:
Apellidos:
Teléfono:
Dirección:

Tipo de contrato: T1 ▼

Cargo a ejercer:
Sueldo base:
Sucursal:
Fecha de inicio:

Guardar Nuevo

Figura 16 Pantalla – Registrar contrato

Si en el menú principal se presiona el botón 'Actualizar contrato' se desplegará la pantalla para modificar contratos existentes, como se ve en la figura 17, el propósito de esta pantalla es modificar información de los contratos, es decir, que al momento de presionar el botón 'Guardar cambios' se sobrescriba la colección de datos existente en los campos correctos sin afectar a los demás, pero solo información realmente modificable, es decir, campos como nombres, apellidos, cédula, estarán bloqueados solo disponibles para visualización del contrato que se está modificando. Si existiera el caso de tener que modificar dichos campos debería existir un módulo independiente en el sistema ERP al cual solo ingresen usuarios de alto nivel con la capacidad de modificar información crítica, esto con el objetivo de evitar errores graves o posibles fraudes.

Figura 17 Pantalla – Actualizar contrato

La siguiente pantalla comprende el núcleo del módulo desarrollado pues aquí se van a generar los roles de pagos de cada empleado mes a mes, para ingresar en la pantalla se debe presionar el botón 'Rol de pagos', desplegada la interfaz gráfica, como se ve en la figura 18, se contempla un buscador el cual requiere el parámetro cedula para funcionar, al ingresar el dato correctamente y existiendo un contrato asociado a la cedula ingresada se visualizar información parcial pero relevante del contrato, como: nombres, apellidos, tipo de contrato y cargo, posteriormente el usuario puede registrar información con la que se hará el cálculo

correspondiente para asignar el sueldo a pagar, para ello se requiere ingresa el número de horas trabajadas al mes, que tendrán un valor diferente por tipo de contrato, también se registra el número de horas extras en caso de poseer el empleado, al igual que bonos o descuento y finalmente una fecha de creación, validada para no repetirse ni ser anterior a la fecha de inicio de contrato. Ingresada toda la información correctamente se presiona el botón 'Calcular' para

desplegar en pantalla el valor del sueldo a pagar del empleado si todo esta correcto se procede a presionar el botón 'Guardar' para almacenar la información en la colección de datos.

LOGO

Generar rol de pagos

Buscar... 🔍

Cédula: XXXXXXXX

Nombres: XXXXXXXX

Apellidos: XXXXXXXX

Tipo de contrato: xx

Cargo: XXXXXXXX

Horas trabajadas:

Hotas extras:

Bonos:

Descuentos:

Fecha:

Sueldo a pagar:

Guardar Calcular Nuevo

Figura 18 Pantalla – Rol de pagos

La última opción del menú principal pero penúltima pantalla es la de 'Buscar rol de pagos', al presionar dicho botón se desplegará una pantalla, como se muestra en la figura 19, que listará los roles de pagos previamente creados a manera de bloques mostrando información resumida, en caso de que se deseará buscar de un empleado en concreto la pantalla cuenta con un buscador por medio de parámetro cedula. Finalizada la búsqueda o si el rol de pagos está a simple vista se puede seleccionar desplegando una nueva pantalla, figura 20, en la cual se puede visualizar de manera detallada el rol de pagos seleccionado.

De esta manera se finaliza el proceso de ejecución del módulo rol de pagos, para regresar al menú principal se puede usar lo botones de navegación.

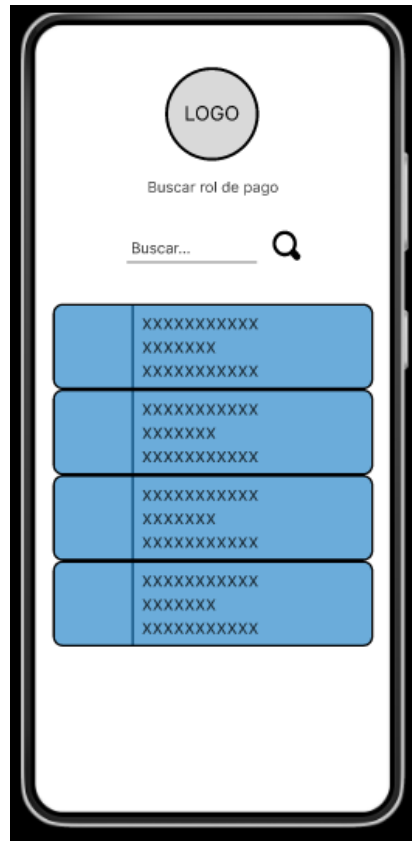


Figura 19 Pantalla – Buscar rol de pagos



Figura 20 Pantalla – Información rol de pagos

2.3.2.2 Estructura de la colección de datos

Con lo mencionado en la fase de requerimientos y mientras se diseñaba las interfaces graficas juntos con las reglas de negocio, se pudo aclarar y establecer la información necesaria que se utilizará en el aplicativo móvil para su correcto funcionamiento.

Es importante esclarecer el porqué del uso de una colección de datos y no de una base de datos relacional. Al trabajar en un proyecto destinado a una aplicación móvil, el manejo de la información almacenada o consumida tiene una mayor facilidad por medio de la herramienta Firebase, explicada en la sección 1.5.2, ya que cuenta con

un vínculo con el editor de código Android Studio, ofreciendo una configuración simple y resultados óptimos.

A demás el módulo rol de pagos no requiere un base de datos muy compleja o estructurada en sus relaciones como se puede ver en la figura 21, se implementó 3 colecciones, una para la información de los contratos, otra para los roles de pagos generados y una tercera muy pequeña almacenando los tipos de contrato, brindando la posibilidad futura de agregar o modificar los tipos de contrato que ofrezca la empresa.



Figura 21 Estructura de colección de datos

2.3.3 Codificación y pruebas de funcionalidad

Ahora que los requerimientos fueron definidos y son claros, se estableció el diseño del aplicativo y se estructuró la colección de datos se pudo proceder a la codificación del producto, para lo cual se empleara la metodología Scrum, previamente explicada en la sección 1.4.3.

Aquí se ejecutaron los eventos Scrum definidos en la sección 2.1 teóricamente, los cuales son Sprint, Sprint Planning, Sprint Review.

2.3.3.1 Sprint 01

Durante el primer Sprint el proyecto se centró en la instalación y configuración de las herramientas necesarias para el desarrollo de producto, al igual que se centró en la primera etapa de recolección de requerimientos por medio de las historias de usuarios.

Sprint Planning

Objetivos:

- Recolección y análisis de requerimientos.
- Instalación y configuración de herramientas.
- Configuración del DevOps para la realización de integración continua durante la codificación.

Ejecución del Sprint

En esta fase inicial del proyecto la atención se enfocó en el entendimiento del producto solicitado con el objetivo de extraer la mayor cantidad de requerimientos y construir una imagen inicial considerando que en etapas futuras esta imagen crecería o cambiaría.

De igual manera durante el periodo de duración del sprint se investigó e implemento las configuraciones iniciales del DevOps, junto con las herramientas descritas en el marco teórico con el propósito de establecer la conexión de Android Studio con Firebase y GitLab con GitKraken buscando conocer su funcionamiento para evitar errores, retrasos o retrabajos.

Sprint Review 01

Tabla 8 Resultados Sprint review 01

Pregunta	Resultado
¿Qué hice en el sprint actual?	Recolección y análisis de requerimiento. Instalación y configuración de las herramientas Android Studio, Firebase, Git, GitKraken, GitLab. Implementación de configuraciones iniciales del DevOps.
¿Qué problemas tuve?	Conflictos al configurar CI/CD por lo que no se pudo completar al 100%
¿Qué hare el siguiente Sprint?	Configurar CI/CD con servicio runner ofrecido por la misma herramienta GitLab. Modelado de interfaces gráficas y colección de datos.

Resultados Sprint 01

La tabla 9, muestra un resumen de los resultados obtenidos durante el primer Sprint.

Tabla 9 Resultados del Sprint 01

Actividad	Ejecución
Recopilación historia de usuario 01	Sí
Recopilación historia de usuario 02	Sí
Recopilación historia de usuario 03	Sí
Recopilación historia de usuario 04	Sí
Preparación de repositorio en GitLab	Sí
Preparación de herramienta Android Studio	si
Preparación de herramienta Firebase	Sí
Preparación de herramienta GitKraken	si
Preparación DevOps	No
Porcentaje obtenido	90%

2.3.3.2 Sprint 02

Sprint Planning

Objetivos:

- Diseñas las interfaces graficas del aplicativo móvil.
- Establecer la colección de datos a utilizar durante el desarrollo y vida del

aplicativo.

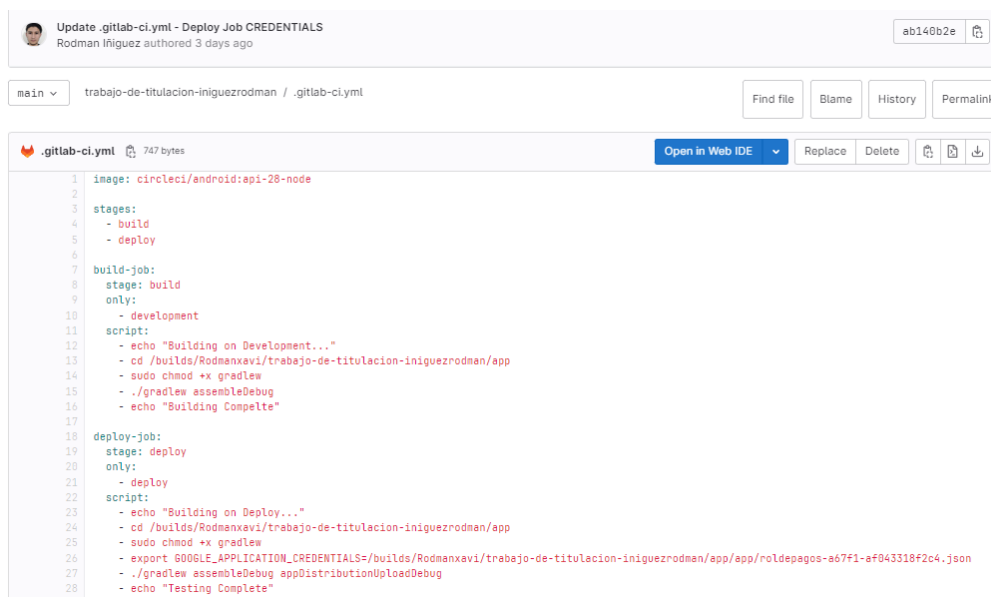
- Culminar la configuración DevOps faltante del primer Sprint.

Ejecución del Sprint

Durante el periodo del segundo Sprint se diseñó en totalidad las pantallas que dispondrá el aplicativo móvil, figura 22, al igual que a lógica de negocio para su funcionamiento, lo cual revelo algunos requerimientos nuevos.

Ya con un diseño visual y con el entendimiento de la lógica de negocio se estructuro la colección de datos necesaria a implementar, figura 22

Otro punto importante logrado durante el segundo Sprint fue la implementación del archivo YML, figura 22 el cual ejecuta el flujo DevOps y efectuar la integración continua con lo cual finalmente se pudo completar la configuración DevOps, como se muestra en la figura 23, justo a tiempo para comenzar la fase de codificación.



```
1 image: circlerci/android:api-28-node
2
3 stages:
4   - build
5   - deploy
6
7 build-job:
8   stage: build
9   only:
10    - development
11   script:
12     - echo "Building on Development..."
13     - cd /builds/Rodmanxavi/trabajo-de-titulacion-iniguezrodman/app
14     - sudo chmod +x gradlew
15     - ./gradlew assembleDebug
16     - echo "Building Compelte"
17
18 deploy-job:
19   stage: deploy
20   only:
21    - deploy
22   script:
23     - echo "Building on Deploy..."
24     - cd /builds/Rodmanxavi/trabajo-de-titulacion-iniguezrodman/app
25     - sudo chmod +x gradlew
26     - export GOOGLE_APPLICATION_CREDENTIALS=/builds/Rodmanxavi/trabajo-de-titulacion-iniguezrodman/app/app/roldepagos-a67f1-af043318f2c4.json
27     - ./gradlew assembleDebug appDistributionUploadDebug
28     - echo "Testing Complete"
```

Figura 22 Codificación archivo yml

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine.

How do runners pick up jobs?

Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure shared runners only handle the jobs they are equipped to run. [Learn more.](#)

Project runners

These runners are assigned to this project.

Set up a project runner for a project

1. Install GitLab Runner and ensure it's running.
2. Register the runner with this URL:
<https://gitlab.com/>

And this registration token:
 GR1348941Pnsfg1N-Pg-5ZRKSz-gQ

Shared runners

These runners are available to all groups and projects.

Each CI/CD job runs on a separate, isolated virtual machine.

Enable shared runners for this project

Available shared runners: 50

- #1506020 (Hs8mheX51)
 windows-shared-runners-manager-1
 shared-windows windows windows-1809
- #1506021 (6QgxEPvRr)
 windows-shared-runners-manager-2

Figura 23 Implementación servicio runner de GitLab

Sprint Review 02

Tabla 10 Resultados Sprint review 02

Pregunta	Resultado
¿Qué hice en el sprint actual?	Se diseño en totalidad las interfaces graficas del aplicativo. Se estructuro la colección de datos empleada durante el proceso de ciclo de vida y posterior. Se completo configuración DevOps.
¿Qué problemas tuve?	Problemas para vicular GitKraken con GitLab
¿Qué hare el siguiente Sprint?	Desarrollo de la aplicación: pantallas de Inicio, Menú principal, Nuevo contrato y Actualizar contrato, con su respectivo funcionamiento.

Resultados Sprint 02

La tabla 11 muestra un resumen de los resultados obtenidos durante el segundo Sprint.

Tabla 11 Resultados del Sprint 02

Actividad	Ejecución
Diseño de la pantalla Inicio.	Sí
Diseño de la pantalla Menú principal.	Sí
Diseño de la pantalla Nuevo contrato.	si
Diseño de la pantalla Actualizar contrato.	Sí
Diseño de la pantalla Rol de pagos.	si
Diseño de la pantalla Buscar rol de pagos.	si
Diseño de la pantalla visualizador rol de pago detallado.	si
Recopilación de nuevos requerimientos.	si
Diseño de la estructura de la colección de datos.	si
Configuración DevOps	si
Conexión de repositorio local con repositorio remoto	no
Porcentaje aproximado:	90%

2.3.3.3 Sprint 03

Sprint Planning

Objetivos:

- Vinculación entre repositorios local y remoto.
- Desarrollo de las Interfaces gráficas de la pantalla de Inicio, Menú principal, Nuevo contrato, Actualizar contrato.

Ejecución del Sprint

Finalmente se pudo conectar los repositorios local y remoto, cambiando la configuración del repositorio remoto de estado privado a público, facilitando a la herramienta hacer el vínculo y poder efectuar las acciones de Push, Pull y Commit, como se muestra en la figura 24 y la figura 25.

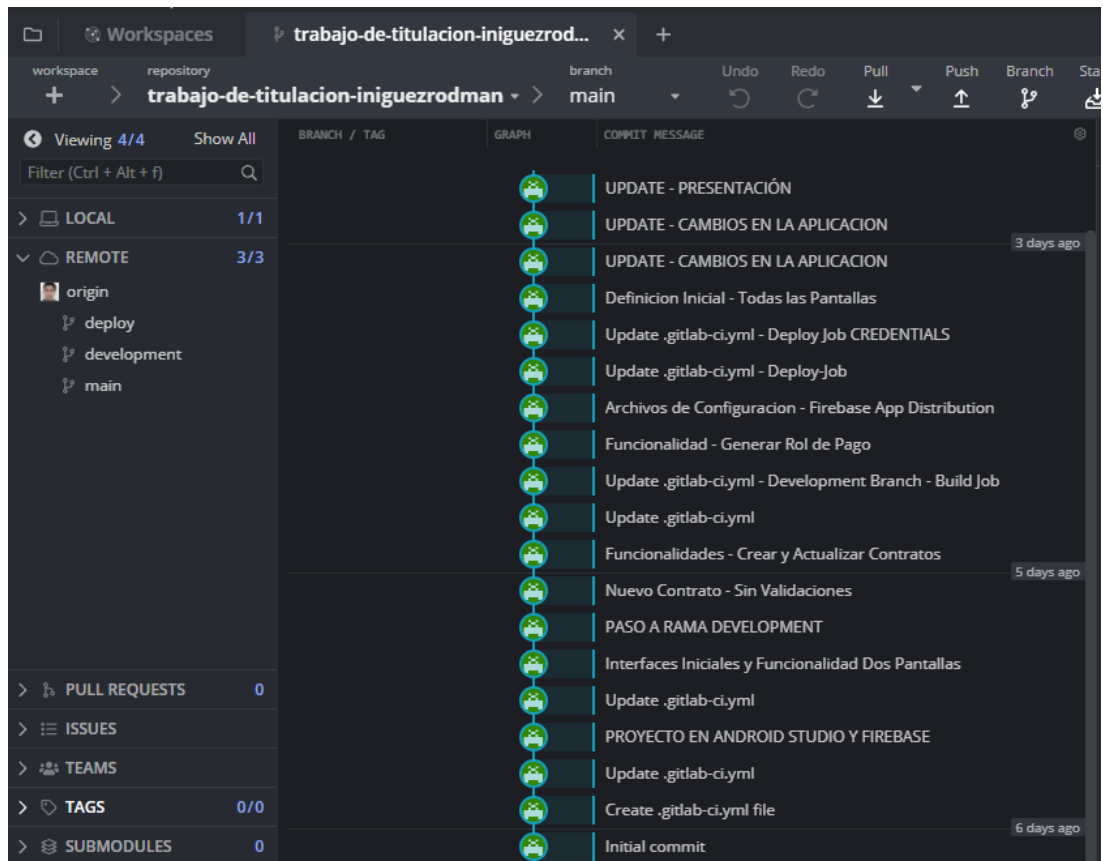


Figura 24 Commits GitKraken

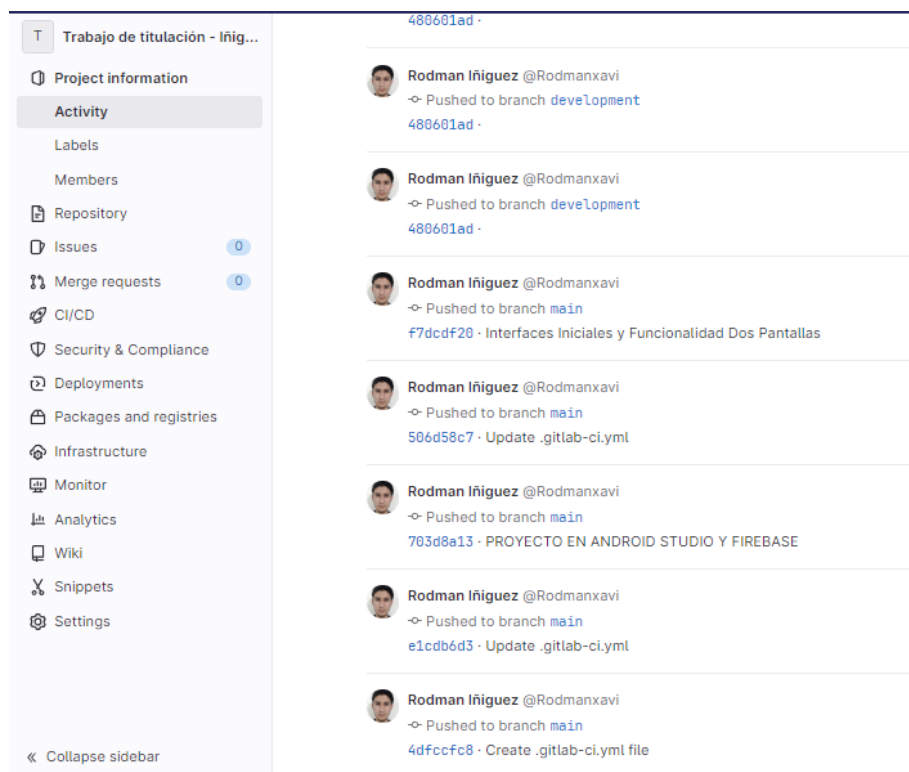


Figura 25 Commits GitLab

Ya con el flujo DevOps ejecutándose como se describe en la sección 2.2.1. Se prosiguió con el desarrollo del aplicativo móvil, empezando por las interfaces gráficas, la primer y más sencilla de todas fue la pantalla de inicio, como se muestra en la figura 26, ya que esta pantalla solo posee un botón el cual da pie de inicio para la ejecución del aplicativo, esta pantalla fue diseñada de esa manera con la intención de poder integrar el módulo rol de pagos al sistema ERP completo.



Figura 26 Implementación pantalla - inicio

Siguiendo con el desarrollo del aplicativo se procedió a codificar la pantalla del menú principal donde se mostrará las opciones de procesos con los que cuenta el usuario para la generación de roles de pago, dichas opciones son:

- Registrar contrato
- Actualizar contrato
- Rol de pagos
- Buscar rol de pagos

Como se muestra en la figura 27



Figura 27 Implementación pantalla - menú principal

Ya marcados los procesos con los que contará el aplicativo móvil en el menú principal se procedió a codificar la pantalla de registro de contrato, la cual permite registrar los datos personales del empleado y esenciales del contrato laboral, dichos datos son: cédula, nombre, apellidos, teléfono, dirección, tipo de contrato, cargo, sueldo base, sucursal y fecha de inicio, todos los datos serán validados al formato correspondiente para evitar errores en el almacenamiento de información crítica como por ejemplo la cedula, nombres y apellidos siendo información que nunca va a cambiar. Finalmente se contará con dos botones, uno para limpiar los campos y otro para guardar la información en la colección de datos. La pantalla se puede visualizar en la figura 28.



REGISTRO DE NUEVO CONTRATO

Cédula:	Ej: 0000000000
Nombres:	Ej: Juan Alberto
Apellidos:	Ej: Pérez Guerrero
Teléfono:	Ej: 0999999999
Dirección:	Ej: Av. Simón Bolívar
Contrato:	Ej: T1
Cargo:	Ej: Asistente Técnico
Sueldo Base:	Ej: 1000000.00
Sucursal:	Ej: La Floresta
Fecha de Inicio:	01/01/1900

Guardar

Limpiar

Figura 28 Implementación pantalla - registrar contrato

Finalizando el tercer sprint, se encuentra la codificación de la ventana y proceso para actualizar un contrato, en esta pantalla se puede buscar un contrato por número de cédula se desplegará la información del empleado, pero no toda será modificable, como se explicó en la pantalla de registrar contrato, con el objetivo de modificar puntos relevantes del contrato que ameriten, más no información crítica del empleado, tal como se muestra en la figura 29



ACTUALIZACIÓN DE CONTRATO

Cédula: Ej: 0000000000

Nombres:

Apellidos:

Teléfono: Ej: 0999999999

Dirección: Ej: Av. Simón Bolívar

Contrato: Ej: T1

Cargo: Ej: Asistente Técnico

Sueldo Base: Ej: 1000000.00

Sucursal: Ej: La Floresta

Fecha de Inicio: 01/01/1900

Guardar Cambios

Nueva Búsqueda

Figura 29 Implementación pantalla - actualizar contrato

Ya realizado el primer avance de codificación se procedió a realizar una prueba del flujo DevOps y poder observar el resultado de dicho flujo en el CI/CD de GitLab, tal como muestra la figura 30, mostrando un resultado favorable comprobando que la implementación DevOps funciona.

passed 00:08:38 3 days ago	Definición Inicial - Todas las Pantallas #775414114 deploy → 69f443db			
passed 00:08:53 3 days ago	Update .gitlab-ci.yml - Deploy Job CREDENTIALS #775293084 deploy → eb146b2e			
failed 00:08:20 3 days ago	Update .gitlab-ci.yml - Deploy-Job #775293084 deploy → 580bc76a			
passed 00:02:32 5 days ago	PASO A RAMA DEVELOPMENT #774507021 deploy → 30bc2eb6			

Figura 30 Ejecución de flujo DevOps

Sprint Review 03

Tabla 12 Resultados Sprint review 03

Pregunta	Resultado
¿Qué hice en el Sprint actual?	Finalmente se logró a conexión entre la herramienta GitKraken y la plataforma GitLab, modificando el proyecto de privado a público. Se desarrollaron todas las interfaces propuestas para este sprint. Se ejecuto una prueba de flujo DevOps con éxito.
¿Qué problemas tuve?	Durante las pruebas de flujo DevOps una no se ejecutó con éxito.
¿Qué hare el siguiente Sprint?	Desarrollo de las del resto de las interfaces gráficas con su funcionamiento, las cuales con generar rol de pagos y visualizar dichos roles de pagos.

Resultados del Sprint 03

La tabla 13, muestra un resumen de los resultados que se obtuvieron en el tercer Sprint.

Tabla 13 Resultados Sprint 03

Actividad	Ejecución
Integración GitKraken con GitLab	Sí
Implementación interfaz gráfica 'Inicio'	Sí
Funcionamiento pantalla Inicio	Si
Implementación interfaz gráfica 'Menú principal'	Sí
Funcionamiento pantalla Menú principal	Sí
Creación colección de datos para contratos	Sí
Implementación interfaz gráfica 'Registrar contrato'	Sí
Funcionamiento pantalla registro contrato	Sí
Implementación interfaz gráfica 'Actualizar contrato'	Sí
Funcionamiento pantalla actualizar contrato	Sí
Prueba de ejecución del flujo DevOps	Sí
Porcentaje aproximado:	100%

2.3.3.4 Sprint 04

Sprint Planning

Objetivos:

- Desarrollo de las interfaces graficas de la aplicación, para generar rol de pagos y visualizar a los mismo.

Ejecución del Sprint

Este cuarto Sprint comprende la segunda parte de codificación para la entrega de un incremento con valor, y claramente será un valor muy alto al completar esta fase, aquí se desarrolló la pantalla, lógica y calculo para generar roles de pagos a partir de contratos ya registrados en la colección de datos y poder visualizarlos de manera resumida y detallada.

Primeramente, se codificó la pantalla para generar los roles de pagos, en esta pantalla se puede encontrar un buscador que requiere el parámetro cedula para buscar un trato existente, de ser así se habilitara los campos para registrar las horas trabajadas, horas extras, bonos y descuentos que posea el empleado junto con la fecha de creación del rol de pagos. Los campos estarán validados con el formato indicado, al igual que la fecha par que no sea ingresada una fecha anterior al inicio del contrato o que se repita un rol de pago dos veces en un mes. Finalmente, para realizar el cálculo del sueldo a pagar se toma en cuenta horas la información ingresada y el tipo de contrato el cual indica el valor por hora trabajada, si los valores obtenidos no tienen ningún error se procede a guardar el rol de pagos generado, caso contrario se tiene la oportunidad de modificar los campos con libertar o buscar un contrato diferente si se necesitará.

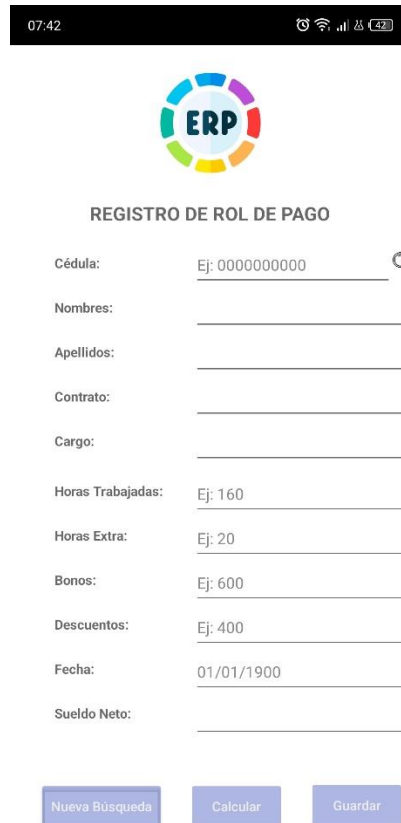
Se menciono que los contratos influyen en el cálculo, eso es debido a la colección tipo de contrato posee valores diferentes para cada uno de ellos los cuales se dividen de la siguiente manera:

- T1 \$2.5 x hora
- T2 \$5 x hora
- T3 \$7.5 x hora
- T4 \$10 x hora

Las horas extras tiene el mismo valor para todo que es de \$4 x hora, si se desea agregar, cambiar o quitar tipos de contratos se puede hacer fácilmente

actualizando la colección que tiene a los tipos de contrato, de igual manera se puede cambiar el valor de la hora y hora extra trabajada para ajustar según la empresa vea necesario.

En la figura 31 se puede observar la pantalla para generar roles de pagos.



07:42

ERP

REGISTRO DE ROL DE PAGO

Cédula: Ej: 0000000000

Nombres: _____

Apellidos: _____

Contrato: _____

Cargo: _____

Horas Trabajadas: Ej: 160

Horas Extra: Ej: 20

Bonos: Ej: 600

Descuentos: Ej: 400

Fecha: 01/01/1900

Sueldo Neto: _____

Nueva Búsqueda Calcular Guardar

Figura 31 Implementación pantalla - rol de pagos

Ya generados los roles de pagos correspondiente a un contrato de trabajo es imperativo el poder visualizarlos para lo cual se diseñaron dos pantallas una que permite buscar los roles de pago por el parámetro cedula de un empleado desplegando a manera de bloques con información resumida, como se indica en la figura 32 y otra pantalla que se abrirá tras seleccionar el rol de pagos deseado para poder visualizar de manera detallada, como se indica en la figura. 33



BUSCAR ROLES DE PAGO

Cédula: 

ROLES DE PAGO:

ROL DE PAGO	
Nombre:	Diana Ines Tipan Cango
Cargo:	Gerente Sucursal
Cédula:	1725169203 Fecha: 13/04/2023
ROL DE PAGO	
Nombre:	Diana Ines Tipan Cango
Cargo:	Supervisora
Cédula:	1725169203 Fecha: 25/03/2023
ROL DE PAGO	
Nombre:	Jose Alejandro Sosa Narvaez
Cargo:	Personal de Aseo
Cédula:	0706856747 Fecha: 25/03/2023
ROL DE PAGO	
Nombre:	Diana Ines Tipan Cango
Cargo:	

Figura 32 Implementación pantalla - buscar rol de pagos



INFORMACIÓN DEL ROL DE PAGO

Cédula:

Nombres:

Apellidos:

Contrato:

Cargo:

Bonos:

Descuentos:

Sueldo:

Fecha:

Figura 33 Implementación pantalla - ver rol de pagos

Sprint Review 04

Tabla 14 Resultados Sprint review 04

Pregunta	Resultado
¿Qué hice en el Sprint actual?	Se implementó la interfaz gráfica y funcionalidad de las pantallas planteadas en el Sprint planning.
¿Qué problemas tuve?	Se presentaron dificultades al momento de establecer cuáles serían los cálculos correspondientes y valores apreciables para el mismo
¿Qué hare el siguiente Sprint?	Correcciones de errores, corrección de validaciones. Lanzamiento del aplicativo

Resultados Sprint 04

La tabla 15, muestra un resumen de los resultados que se obtuvieron en el cuarto Sprint.

Tabla 15 Resultados Sprint 04

Actividad	Ejecución
Implementación interfaz gráfica 'Generar rol de pagos'	Sí
Funcionamiento pantalla generar rol de pagos	Sí
Implementación interfaz gráfica 'Buscar rol de pagos'	Sí
Funcionamiento pantalla buscar rol de pagos	Sí
Implementación interfaz gráfica 'Ver rol pagos'	Sí
Funcionamiento pantalla ver rol de pagos	Sí
Porcentaje aproximado:	100%

2.3.3.5 Sprint 05

Sprint Planning

Objetivos:

- Corrección de errores de validación y diseño.
- Lanzamiento final del aplicativo móvil.

Ejecución del Sprint

Este último Sprint se enfocó en corrección de errores pequeños como diseño, validaciones, colores y principalmente en pruebas de flujo DevOps como se evidencia en la figura 34, de igual manera se realizó el lanzamiento final del aplicativo movil haciendo uso de la herramienta App-Distribution que se encuentra en la plataforma Firebase, figura 35.

Status	Pipeline	Triggerer	Stages
passed 00:08:35 2 days ago	UPDATE FINAL - (2) #776669581 depLoy -> 680d154f latest	[Avatar]	✓
passed 00:09:05 2 days ago	UPDATE FINAL - (1) #776621380 depLoy -> 5f286378	[Avatar]	✓
passed 00:08:46 2 days ago	UPDATE - PRESENTACION #776589087 depLoy -> 84d37110	[Avatar]	✓
passed 00:08:58 2 days ago	UPDATE - CAMBIOS EN LA APLICACION #776583454 depLoy -> 6e4a153c	[Avatar]	✓
passed 00:07:56 3 days ago	UPDATE - CAMBIOS EN LA APLICACION #775469807 depLoy -> 3f5f1089	[Avatar]	✓
passed	Definición Inicial - Todas las Pantallas	[Avatar]	✓

Figura 34 Ejecución de flujo DevOps final



Figura 35 Lanzamiento del producto por App-Distribution

Sprint Review 05

Tabla 16 Resultados Sprint review 05

Pregunta	Resultado
¿Qué hice en el Sprint actual?	Pruebas finales del flujo DevOps. Correcciones finales y lanzamiento del aplicativo móvil.
¿Qué problemas tuve?	No se presentó dificultades
¿Qué hare el siguiente Sprint?	El presente es el último Sprint

Resultados Sprint 05

La tabla 17, muestra un resumen de los resultados obtenidos en el quinto y último Sprint.

Tabla 17 Resultados Sprint 05

Actividad	Ejecución
Correcciones de errores de diseño	Sí
Validación de datos	Sí
Pruebas finales del flujo DevOps	Sí
Lanzamiento final de aplicativo móvil	Sí
Porcentaje aproximado:	100%

2.3.4 Pruebas de Usabilidad

Evaluación de Usabilidad

Ya finalizado todo el proceso de implementación llego el momento de instaurar pruebas al aplicativo móvil, al igual que todo software creado, debe cumplir con niveles de usabilidad que justifiquen la creación de este.

El sistema para medir la usabilidad del aplicativo movil desarrollado será por medio de Escala de Usabilidad del Sistemas o SUS (System Usability Scale) [20], este método consta de un cuestionario que contiene 10 preguntas cuyo objetivo es obtener información de la experiencia que tuvieron diferentes usuarios al momento de usar el producto. Estas preguntas, se muestran en la tabla 18, se medirán en una escala que va del 1 al 5 que la final nos permitirá dar un porcentaje de aprobación o rechazo sobre la aplicación acerca de su uso.

Tabla 18 Preguntas SUS [20]

# Pregunta	Pregunta (Sentencia)
1	Pienso que me gustaría utilizar este sistema frecuentemente
2	Encuentro el sistema innecesariamente complicado
3	Pienso que el sistema fue fácil de usar
4	Pienso que necesitaría ayuda de un técnico para poder utilizar el sistema
5	Pienso que las funciones del sistema fueron bien integradas
6	Pienso que existe mucha inconsistencia dentro del sistema
7	Pienso que la mayoría de las personas aprenderían a utilizar este sistema muy rápido
8	Encontré el sistema muy complicado de usar
9	Sentí mucha confianza al utilizar el sistema
10	Necesité aprender muchas cosas antes de usar el sistema

Ya teniendo claro con que método se evaluara la usabilidad y la forma de puntuar cada pregunta, es importante tener lo siguiente en consideración ya que forma

parte vital del funcionamiento de SUS:

- El puntaje colocado a las preguntas impar, se le resta en 1.
- El puntaje colocado a las preguntas par, se resta dicho puntaje de 5.
- Una vez hechos los cálculos mencionados se procede a sumar los resultados.
- Para finalmente multiplicar el valor resultante por 2.5, se multiplica dicho valor por 2.5, provocando que las puntuaciones finalmente obtenidas se encuentren en un rango comprendido entre 40 y 100 [20].

De tal manera que se puede instaurar una tabla 19 describiendo los resultados obtenidos y comprender de mejor manera el nivel de usabilidad el producto desarrollado en base a una escala.

Tabla 19 Resultados SUS

Rangos de valores	Clasificación del resultado
40.00 – 55.00	Deficiente
55.01 – 70.00	Regular
70.01 – 75.00	Buena
75.01 – 100.00	Sobresaliente

Explicado que es el método SUS y cómo funciona luego el momento de aplicarlo al proyecto. El objetivo claramente medir el nivel de usabilidad con el usuario al igual que recopilar todo comentario, opinión o crítica que pueda mejorar al proyecto para un futuro.

Para la ejecución del método SUS se procedió de la siguiente manera, se creó un cuestionario con la herramienta Microsoft Forms implementando las 10 preguntas mostradas en la tabla 18, luego se solicitó a varios participantes probar la aplicación ofreciéndoles el apk y luego de usar la aplicación llenar la encuesta, figura 36.

Encuesta de Usabilidad del Sistema (Contabilidad)

El siguiente cuestionario presenta 10 aspectos relativos al uso de la aplicación presentada al usuario, a través de los cuales se evaluará la experiencia del usuario durante el uso de dicha aplicación.

* Obligatorio

1. Encuesta de Usabilidad del Sistema *

	Muy en desacuerdo	En desacuerdo	Ni en acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo
Pienso que me gustaría utilizar este sistema frecuentemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encuentro el sistema innecesariamente complicado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 36 Cuestionario SUS

Finalizado todo el proceso de pruebas por parte de los usuarios llegó el momento de calificar, para lo cual la tabla 20 nos servirá para asignar el valor correspondiente a las respuestas seleccionadas por parte de los usuarios.

Tabla 20 Equivalencia SUS

Descripción	Escala (1 a 5)
Muy en desacuerdo	1
En desacuerdo	2
Ni en acuerdo ni en desacuerdo	3
De acuerdo	4
Muy de acuerdo	5

Resultado pruebas de usabilidad

Ya con las encuestas resueltas por parte de los usuarios y conociendo el método de calificación, explicado al inicio de la sección, se colocó los resultados en tablas para una mejor visualización de estos.

Tabla 21 Respuestas usuario 01

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	4
2	1
3	4
4	2
5	4
6	1
7	5
8	1
9	4
10	3
Puntaje Total:	82.50/100.00

Tabla 22 Respuestas usuario 02

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	2
3	4
4	3
5	5
6	2
7	5
8	1
9	5
10	1
Puntaje Total:	87.50/100.00

Tabla 23 Respuestas usuario 03

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	1
3	5
4	1
5	5
6	1
7	5
8	1
9	5
10	2
Puntaje Total:	97.50/100.00

Tabla 24 Respuestas usuario 04

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	4
3	5
4	2
5	3
6	2
7	4
8	2
9	4
10	1
Puntaje Total:	75.00/100.00

Tabla 25 Respuestas usuario 05

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	1
3	4
4	1
5	5
6	2
7	3
8	1
9	4
10	2
Puntaje Total:	85.00/100.00

Tabla 26 Respuestas usuario 06

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	4
2	2
3	4
4	3
5	4
6	1
7	3
8	1
9	5
10	4
Puntaje Total:	72.50/100.00

Tabla 27 Respuestas usuario 07

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	1
3	4
4	2
5	5
6	2
7	4
8	3
9	5
10	3
Puntaje Total:	80.00/100.00

Tabla 28 Respuestas usuario 08

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	2
3	5
4	2
5	5
6	2
7	5
8	2
9	5
10	2
Puntaje Total:	87.50/100.00

Tabla 29 Respuestas usuario 09

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	2
3	3
4	3
5	3
6	1
7	4
8	1
9	5
10	3
Puntaje Total:	75.00/100.00

Tabla 30 Respuestas usuario 10

Número de Pregunta	Respuesta (Escala del 1 al 5)
1	5
2	1
3	4
4	1
5	4
6	1
7	4
8	2
9	5
10	3
Puntaje Total:	77.50/100.00

2.3.5 Despliegue

Después de todo un proceso complejo, investigando y utilizando nuevas herramientas de desarrollo, nuevas configuraciones de los métodos empleados, horas de diseño y codificación de someter los flujos de trabajo a pruebas al igual que el producto final a pruebas de usabilidad llega el momento del lanzamiento del aplicativo móvil el cual por medio de la plataforma Firebase, utilizando App-Distribución, como indica la figura 37, se presentan facilidades y ventajas para su distribución como un plus agregado al método DevOps planteado desde el inicio del proyecto, obteniendo un archivo APK con todas las funcionalidad esperadas para el módulo rol de pagos.



Figura 37 App-Distribution lanzamiento del producto final

Por último, y no menos importante con el producto ya liberado se consideró la construcción de un manual de usuario, el cual está disponible en el Anexo, para una fácil comprensión del funcionamiento del aplicativo al igual que en el Anexo se encontrará videos demostrativos.

3 RESULTADOS

Tras el arduo trabajo que significo el desarrollo de este proyecto, es momento de presentar los resultados obtenidos, partiendo de nuestro objetivo principal el cual consta de una comparativa entre el desarrollo de un aplicativo móvil haciendo uso de buenas prácticas DevOps y el desarrollo del aplicativo móvil sin buenas prácticas DevOps.

Para ellos se encontraron ciertos aspectos que los llamaremos métricas, con el objetivo de poder analizar y cuantificar los procesos efectuados, dichas métricas se basan en los flujos DevOps y estas son:

- Pipelines aprobados
- Pipelines fallidos
- Pipelines Cancelados

En la tabla 31 se muestran cuantificados y clasificados los resultados de los 29 Pipelines ejecutados durante el ciclo de vida del proyecto.

Tabla 31 Resultados pipelines

Ramas	Estado del Flujo	Flujos
Development	Exitoso	18
	Fallido	1
	Cancelado	1
Deploy	Exitoso	8
	Fallido	1
	Cancelado	0

La alta cantidad de flujos exitosos contra la baja cantidad de flujos fallidos o cancelados evidenciados en la tabla 31, son el producto de dos factores principales, el primero fue, y gracias a, a una gran consulta de información y ligeras entrevistas con personas experimentada en métodos DevOps que brindaron concejos útiles, por lo que se pudo realizar una implementación casi limpia desde el comienzo justificado por el único caso fallido para cada una de las ramas, y el segundo factor fue gracias a la utilización de Runners brindados

por la propia plataforma GitLab lo que evitaba gran cantidad de conflictos como sería con otras herramientas, por ejemplo Docker debido a su compleja configuración.

Analizando los flujos ejecutados y los motivos de esos resultados salta evidentemente que una buena configuración DevOps permite a los equipos de trabajo una realización de una integración continua eficiente del o los proyectos lo que significa optimización en tiempos de desarrollo. Para una mejor comprensión de lo mencionado, la figura 38 indica en porcentaje la cantidad de trabajo realizada con éxito versus los fracasos.

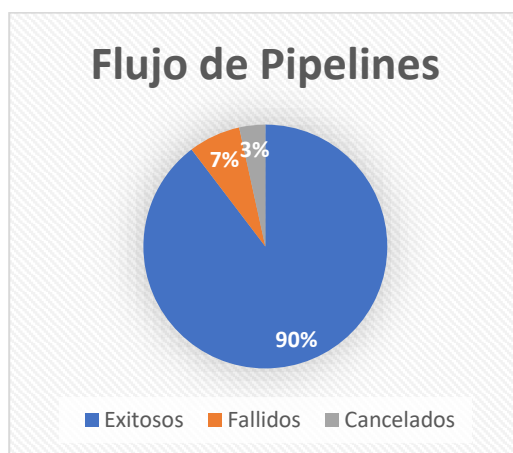


Figura 38 Flujos DevOps ejecutados

Después de observar como el flujo DevOps ayudo a optimizar el tiempo, sin olvidar que el equipo de desarrollo está conformado por un solo integrante y eso reduce tiempos de “comunicación entre equipos” para cada etapa del desarrollo, se vuelve relevante analizar porcentajes logrados en cada Sprint ejecutado. como se muestra en la tabla 32, se evidencia un alto porcentaje de éxito en cada bloque, lo que evita acumulación de trabajo en los Sprint posteriores de los anteriores, lo que podría derivarse como ventaja del método DvOps pero al no probarse con varios equipos no se puede asegurar con fidelidad.

Tabla 32 Porcentajes de éxito por Sprint

Número de Sprint	Porcentaje de avance en relación con la planificación
1	90.00 %
2	90.00 %

3	100.00 %
4	100.00 %
5	100.00 %
Promedio:	96.00 %

Finalmente, las pruebas de usabilidad, estas pruebas pueden dar un mejor vistazo al resultado que se obtuvo aplicando DevOps, en otras palabras, se acorto tiempo de producción y el producto cumple con la funcionalidad más esencial del módulo al igual que otras complejas, pero será suficiente para poder ofrecer una buena aplicación móvil al usuario, para ello la tabla 33 evidencia el resultado con respecto a la usabilidad y si cumple par que el usuario final haga uso prolongado de la misma.

Tabla 33 Resultados SUS

Encuesta	Puntaje Total	Clasificación
1	82.50	Bueno
2	87.5	Sobresaliente
3	97.50	Sobresaliente
4	75.00	Bueno
5	85.00	Regular
6	72.50	Bueno
7	80.00	Bueno
8	85.00	Sobresaliente
9	75.00	Bueno
10	77.50	Bueno
Promedios:	81.75	Bueno

En la tabla 33 como resultado de la prueba usabilidad vemos una aceptación media por parte de los usuarios con un 81.75 que no representa una mala puntuación, pero tampoco es una excelente, se puede deducir en base a las respuestas seleccionadas en cada pregunta que se puede hacer aún más amigable la aplicación para el usuario dedicando un poco más de tiempo a la etapa de diseño y en base a la efectividad de las actividades en cada Sprint es posible aumentar dicha carga de trabajo.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Como objetivo principal del proyecto se estableció una comparativa entre el desarrollo de un aplicativo móvil empleado prácticas DevOps y el mismo desarrollo sin prácticas DevOps, ya aclaro y recalco varias veces en el presente documento, dicho método se basa en la relación de los procesos de desarrollo y operaciones, es decir y resumiendo en pocas palabras una integración continua de entregas funcionales que aporten valor al producto y avances en el proyecto donde todos los equipos de trabajo noten los aportes de los demás como los suyos, apoyados por diferentes herramientas como App-Distribution que facilitaba el lanzamiento de versiones de un APK funcional con cada flujo DevOps si se deseaba. Expuesto de esta manera el método DevOps junto con la metodología Scrum se esperaba un desarrollo más eficaz y rápido, pero aquí surgió el mayor inconveniente, en el momento que solo existió un equipo de desarrollo conformado por una persona los flujos DevOps se los pudo observar funcionar pero tuvieron poca sensación de impacto ya que una sola persona se encarga de ejecutar los flujos, realizar las pruebas y obtener la versión del lanzamiento al flujo correspondiente, así hasta terminar el desarrollo del aplicativo móvil por completo. De esta manera se pudo deducir que en equipos de trabajos mayores donde se realizan diferentes módulos o funcionalidades al mismo tiempo tendría un mayor impacto y se observarían diferentes resultados de cuando no se emplea un flujo DevOps.

Otro punto importante por destacar que salió a la luz durante el desarrollo del proyecto fue el implementar herramientas correctas y una configuración adecuada para los flujos, debido a que una configuración confusa o errónea al igual que el uso de herramientas con un nivel complejo de integración al método DevOps reflejaban un retraso en los avances iniciales hasta que todos los errores sean corregidos y poder realizar los flujos de trabajo de integración continua.

En el planteamiento inicial del proyecto se estableció usar DevOps y Scrum de la mano, peor, hay que hacer notar que Scrum es una metodología muy fuerte que por sí sola ha sacado al mundo muchos y diferentes proyectos, por lo que planteo si era necesario el flujo DevOps durante algunos puntos del ciclo de vida de desarrollo, curiosamente la respuesta es un si y no, explicando un poco más, el método DevOps es muy poderoso pero debe analizarse en qué proyecto de

implementará, de ahí a respuesta de 'sí y no', ya que existirán proyectos donde no se verá un resultado verdadero hasta el final con un único lanzamiento del app, o casos particulares donde el equipo de trabajo es muy reducido que será imperceptible. Pero de igual manera como fue en este caso en particular apoyo mucho para el lanzamiento de versiones funcionales útiles hasta la culminación del producto en desarrollo.

4.2 Recomendaciones

Claramente la mayor recomendación que se puede hacer es implementar el método DevOps en proyectos con diferentes equipos de trabajos de manera que existan varias líneas de ejecución usando el flujo DevOps de tal modo que la integración sea más evidente y apoye en mayor medida al desarrollo del producto.

Del mismo modo, una correcta definición del flujo DevOps es necesaria con fin de realizar un buen desempeño por eso se recomienda estar al pendiente de las actualizaciones de las herramientas que permiten la instauración del método, esta recomendación se basa en un punto particular sucedido durante el desarrollo del proyecto, en el cual se realizó una configuración inicial del flujo DevOps, y por una desafortunada coincidencia, las políticas de las herramientas había cambiado y presento muchos problemas hasta dar con las nueva forma de configuración la cual ya en el momento que se implementó otorgo un flujo de trabajo más estable para un buendesempeño que se evidencio en la sección 3 de resultados.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] (2023). ¿Qué es ERP y por qué se usa en una empresa? | QAD. <https://www.qad.com/es-MX/what-is-erp>
- [2] ¿Qué es DevOps? Explicación de DevOps | Microsoft Azure. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>
- [3] (2019). ¿Qué es DevOps y para qué sirve? | NetApp. <https://www.netapp.com/es/devops-solutions/what-is-devops/>
- [4] K. Schwaber, J. Sutherland, (2020), "The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game," | <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [5] Introducción a Android Studio. | Android Developers. <https://developer.android.com/studio/intro?hl=es-419>
- [6] Información básica. | Firebase. <https://firebase.google.com/docs/guides?hl=es-419>
- [7] (2019). Modelo de datos de Cloud Firestore | Firebase. <https://firebase.google.com/docs/firestore/data-model?hl=es-419>
- [8] (2019). Firebase App Distribution. | Firebase. <https://firebase.google.com/docs/app-distribution?hl=es-419>
- [9] (2023). ¿Qué es figma?, La mejor herramienta de prototipado web. | CEI: Escuela de Diseño y Marketing. <https://cei.es/que-es-figma/>
- [10] (2019). Git - About Version Control. | git-scm. <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- [11] (2019). What is a Git Repository? | Beginner Git Tutorial. | GitKraken. <https://www.gitkraken.com/learn/git/tutorials/what-is-a-git-repository>
- [12] (2019). Git - Branches in a Nutshell. | git-scm. <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>
- [13] (2019). Git - Working with Remotes. | git-scm. <https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes>
- [14] A. (2021). GitKraken. | Alexmarket. <https://alexmarket.medium.com/gitkraken-ea27eb8e8301>
- [15] About GitLab. | GitLab. <https://about.gitlab.com/company/>
- [16] GitLab CI/CD | GitLab. <https://docs.gitlab.com/ee/ci/>
- [17] GitLab Runner | GitLab. <https://docs.gitlab.com/runner/>
- [18] Descripción general de Kotlin. | Android Developers. <https://developer.android.com/kotlin/overview?hl=es-419>

- [19] (2021). ¿Qué es YAML? YAML y su uso en la automatización. | REDHAT. <https://www.redhat.com/es/topics/automation/what-is-yaml>
- [20] C. Busquets. (2021). Medir la usabilidad con el Sistema de Escalas de Usabilidad (SUS). | uiFromMars. <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>

6 ANEXOS

ANEXO I

En el enlace contiene el repositorio con el código fuente de implementación del módulo rol de pagos desarrollado.

<https://gitlab.com/Rodmanxavi/trabajo-de-titulacion-iniguezrodman.git>

ANEXO II

Historia de Usuario #01

Versión: 0.1

Registro de un contrato nuevo

Antecedentes: Yo, como usuario, deseo ingresar al sistema y poder registrar un contrato de un empleado nuevo

- **Escenario:** Visualizar pantalla para registro de un contrato
 - **Cuando** he ingresado a la opción “Registrar contrato”
 - **Entonces** se mostrará una pantalla con todos los campos en blanco para ingresar la información del nuevo empleado.
- **Criterios de aceptación**
 - Registro nuevo contrato en base datos
 - **Campos primarios**
 - Cedula, nombres, apellidos, teléfono, dirección, tipo contrato, cargo, sueldo base, sucursal, fecha de inicio.
 - **Campos secundarios**
 - Ninguno
- **Comentarios** (opcional)
 - Se solicita validación de los campos, formatos correctos en cada uno
 - se solicita un solo contrato asociado a un numero de cedula.

ANEXO III

Historia de Usuario #02

Versión: 0.1

Actualización de un contrato existente

Antecedentes: Yo, como usuario necesito poder actualizar o modificar información de los contratos previamente registrados.

- **Escenario:** Buscar datos de un contrato existente para modificación de información.
 - **Cuando** he ingresado a la opción “Actualizar contrato”
Entonces se mostrará una pantalla para realizar una búsqueda por número de cedula asociado a un contrato y cambiar información de este.
- **Criterios de aceptación**
 - Actualizar información de contrato almacenado en la base de datos.
 - **Campos primarios**
 - Teléfono, dirección, tipo contrato, cargo, sueldo base, sucursal, fecha inicio.
 - **Campos secundarios**
 - Ninguno
- **Comentarios** (opcional)
 - Se solicita visualizar, sin poder modificar campos como cedula, nombres, apellido, al actualizar un contrato.
 - Se solicita verificar formato de campos editables.

ANEXO IV

Historia de Usuario #03

Versión: 0.1

Generar roles de pagos vinculados a un contrato existente

Antecedentes: Yo, como usuario necesito poder generar roles de pagos de los empleados asociados a un contrato mes a mes.

- **Escenario:** Buscar datos del empleado por medio de su cedula para realizar el cálculo del sueldo a pagar en base a su tipo de contrato.
 - **Cuando** he ingresado a la opción “Rol de pagos”
 - **Entonces** se mostrará una pantalla en la cual se pueda buscar datos del empleado y contrato adjuntos para asignar valor a calcular.
- **Criterios de aceptación**
 - Generar rol de pagos de un empleado
 - **Campos primarios**
 - Horas trabajadas, horas extras, bonos, descuentos, fecha, sueldo a pagar
 - **Campos secundarios**
 - Ninguno
- **Comentarios** (opcional)
 - Establecer rango para horas trabajadas y horas extras
 - Verificar que no se repita la fecha de generación del rol de pagos para evitar errores o fraudes
 - Calcular el sueldo a pagar en base al tipo de contrato, cada contrato contara con un valor diferente a pagar por hora trabajada.

ANEXO V

Historia de Usuario #04

Versión: 0.1

Buscar rol de pago de un empleado

Antecedentes: Yo, como usuario necesito poder buscar los roles de pagos de los empleados, obteniendo una vista resumida de la información.

- **Escenario:** Buscar rol de pagos existente en base de datos asociado al número de cedula de un empleado.
 - **Cuando** he ingresado a la opción “Buscar rol de pagos”
Entonces se mostrará una pantalla para realizar una búsqueda por número de cedula asociado a un empleado y visualizar de manera resumida el o los roles de pagos generados a su contrato de trabajo vigente.
- **Criterios de aceptación**
 - Actualizar roles de pagos almacenados en la base de datos.
 - **Campos primarios**
 - Cedula.
 - **Campos secundarios**
 - Ninguno
- **Comentarios** (opcional)
 - Se solicita visualizarlos roles de pagos generados conforme a la fecha.
 - Se solicita filtrar los roles de pagos de un empleado usando el número de cedula.
 - Se solicita ver los roles de pagos con información esencial resumida.

Versión: 0.2

Visualizar rol de pago de un empleado con información detallada

Antecedentes: Yo, como usuario necesito poder visualizar el rol de pagos seleccionado de un empleado con información más detallada.

- **Escenario:** Filtrado el o los roles de pago y mostrado en bloques con información esencial, al seleccionar el rol acceder a una nueva pantalla con información más detallada.
 - **Cuando** he ingresado al rol de pagos a inspeccionar
Entonces se mostrará una pantalla con información detallada del rol de pagos seleccionado.

- **Criterios de aceptación**
 - Visualizar rol de pagos almacenado en la base de datos.
 - **Campos primarios**
 - Ninguno.
 - **Campos secundarios**
 - Ninguno
- **Comentarios** (opcional)
 - Se solicita ver el rol de pagos con información detallada.

ANEXO VI

El enlace contiene el Mockup del aplicativo realizado con la herramienta Figma:

<https://www.figma.com/file/Z1avRmw0iOnk4gc7uFyGXN/Trabajo-titulaci%C3%B3n---Rodman-I%C3%B1iguez?node-id=0%3A1&t=stWQG5WCmBR8Lmi-1>

ANEXO VII

El enlace contiene el diseño de la colección de datos realizado en la plataforma Lucidchart:

https://lucid.app/lucidchart/1a542d5d-814d-44af-9e69-d13cb6760d32/edit?viewport_loc=-282%2C114%2C1682%2C756%2CdDlxMhwKpNSI&invitationId=inv_648e0e50-59a1-45e2-861c-df768d29947e

ANEXO VIII

El enlace contiene un video explicativo del funcionamiento del aplicativo móvil, módulo rol de pagos.

https://epnecuador-my.sharepoint.com/:f/g/personal/rodman_iniguez_epn_edu_ec/EkeuM2iwWjxAkZnN4z4ZICsB0X5ST4wjSpDEAo4EGbBZQA?e=uZLyLT

ANEXO IX

El enlace contiene una carpeta que permite descargar el APK funcional del producto terminado.

https://epnecuador-my.sharepoint.com/:f/g/personal/rodman_iniguez_epn_edu_ec/ErPEM5GQZiVMtct6_hYZShgBVXREIUebpYWeHW7MI-kRvQ?e=CtsrWs